



HAL
open science

Optimizing the computation of overriding in DLN

Piero A. Bonatti, Iliana M. Petrova, Luigi Sauro

► **To cite this version:**

Piero A. Bonatti, Iliana M. Petrova, Luigi Sauro. Optimizing the computation of overriding in DLN. Artificial Intelligence, 2022, 311, pp.103764. 10.1016/j.artint.2022.103764 . hal-03934420

HAL Id: hal-03934420

<https://hal.science/hal-03934420>

Submitted on 11 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Optimizing the computation of overriding in \mathcal{DL}^N

P. A. Bonatti^a, I. M. Petrova^b, L. Sauro^a

^a*Dept. of Electrical Eng. and Information Technologies
Università di Napoli Federico II, Italy*

^b*Inria, Sophia Antipolis, France*

Abstract

One of the factors that is hindering the adoption of nonmonotonic description logics in applications is performance. Even when monotonic and nonmonotonic inferences have the same asymptotic complexity, the implementation of nonmonotonic reasoning may be significantly slower. This happens also with the family of nonmonotonic logics \mathcal{DL}^N .

In this work we address this issue by introducing two optimizations for reasoning in \mathcal{DL}^N . The first optimization, called *optimistic evaluation*, aims at exploiting incremental reasoning in a better way. The second is a *module extractor for \mathcal{DL}^N* , that has the purpose of focusing reasoning on a relevant subset of the knowledge base. The proposed optimization *iterates* the module extractor that, unlike classical module extractors, is not idempotent, in general.

We prove that the proposed optimizations are correct and complete, and assess them through extensive experiments. Our results prove that optimized \mathcal{DL}^N reasoning is often compatible with interactive query answering, which brings nonmonotonic description logics closer to practical applications.

Keywords: Nonmonotonic Reasoning, Description Logics, Module extraction

1. Introduction

In many areas, knowledge is formulated by stating general properties and definitions that are progressively refined by specifying *exceptions*. For example, this approach is extensively used in biology; here we report two representative examples. First, the human heart is usually located in the left-hand half of the thorax. Still, there are exceptional individuals, with so-called *situs inversus*, whose heart is located on the opposite side. Second, eukaryotic cells are those with a proper

nucleus, by definition. Still, they comprise mammalian red blood cells, that in their mature stage have no nucleus.¹

Analogously, the formal languages designed to describe security and privacy policies generally allow to formulate default conditions, such as *open* and *closed* policies,² conflict resolution methods such as *denials take precedence*, and authorization inheritance with exceptions [3].

Nevertheless, description logics (DLs) – which underlie the Semantic Web standard OWL2, and have been proposed as policy languages – do not support defeasible knowledge and exceptions. Consequently, several authors advocated nonmonotonic semantics as a useful means to address this limitation, and proposed different formalisms based on circumscription [4, 5, 6], autoepistemic logic [7, 8], typicality operators [9, 10, 11], or rational closure [12, 13, 14], just to name a few.

In this context, \mathcal{DL}^N [15, 16, 17] is a recent family of nonmonotonic DLs that results from a utilitarian approach to nonmonotonic reasoning. The main goal of this approach is addressing the practical needs of ontology and policy designers, that have been illustrated with several examples in the literature, including the aforementioned papers on biomedical ontologies and semantic web policies. \mathcal{DL}^N features *normality concepts* NC to denote the standard/prototypical instances of a concept C , and prioritized *defeasible inclusions* (DIs) $C \sqsubseteq_n D$ that mean (roughly speaking): “*by default, all prototypical instances that satisfy C satisfy also D , unless stated otherwise*”, that is, unless some higher priority axioms contradict this implication; in this case, $C \sqsubseteq_n D$ is *overridden*. The standard/prototypical instances of C must satisfy all the DIs that are not overridden in C .

\mathcal{DL}^N benefits from some distinguishing features such as: (i) \mathcal{DL}^N adopts the simplest possible criterion for overriding, that is, inconsistency with higher priority axioms; (ii) all the normal instances of a concept C conform to the same set of default properties, sometimes called *prototype*; (iii) the conflicts between DIs that cannot be resolved with priorities can be easily detected and fixed by knowledge engineers; (iv) priorities are not cast into the logic, they are part of the knowledge base. Here is a summary of the main strenghts deriving from (i)–(iv).

No inheritance blocking. Most of the logics grounded on preferential semantics and rational closure block the inheritance of *all* default properties in excep-

¹All of these examples are introduced and discussed in [1, 2].

²If no explicit authorization has been specified for a given access request, then an open policy permits the access while a closed policy denies it.

tional subclasses (as opposed to overriding only the properties that are modified in those subclasses). \mathcal{DL}^N 's simple overriding mechanism does not suffer from this drawback.

No undesired CWA effects. Many nonmonotonic DLs extend default properties to as many individuals as possible, thereby introducing CWA (i.e. closed-world assumption) effects that clash with the intended behavior of ontologies. \mathcal{DL}^N does not introduce any CWA effect because it does not force individuals to be normal, unless explicitly stated otherwise.

Control on priorities. Since priorities are not fixed a priori in \mathcal{DL}^N , knowledge engineers can adapt them to their needs. It is possible to override DIs based on temporal criteria (which may be useful in legal ontologies and ontology versioning, where new assertions partially replace older assertions), or based on specificity (i.e. the properties of more specific classes override the conflicting properties of their superclasses), and other custom conflict resolution criteria. The logics derived from inheritance networks, preferential semantics, and rational closure can only support their fixed, specificity-based overriding criterion.

Default role fillers. \mathcal{DL}^N axioms can specify whether a role should range only over normal individuals or not. Some logics are completely unable to apply default properties to role values.³ Some others cannot switch this inference off when it is not desired. Only \mathcal{DL}^N and $\mathcal{ALC} + \mathbf{T}_{\min}$ make it possible to control this kind of inference.

Inconsistent prototype detection. \mathcal{DL}^N facilitates the identification of all conflicts that cannot be resolved with priorities (via consistency checks over normality concepts), because their correct resolution is application dependent and should require human intervention. Typically, unresolved conflicts correspond to knowledge gaps or errors.

Unique deductive closure. As a result of the automated handling of unresolved conflicts, several nonmonotonic logics yield multiple deductive closures, that correspond to all the possible ways of solving each conflict. \mathcal{DL}^N is one of the logics that has a unique closure.

Generality. \mathcal{DL}^N can be uniformly applied to all description logics up to the standard OWL2-DL (i.e. the logic $\mathcal{SROIQ}(\mathcal{D})$) and beyond. Typicality logics and rational closure, instead, are limited to logics that satisfy the *disjoint union*

³This is the case for rational closure. Recently, in [18], a solution has been proposed for \mathcal{EL} with \perp . It is unclear how to extend it to more expressive DLs, and it is not possible to “turn off” the application of default rules to role fillers.

model property. Recently, it has been shown that for expressive DLs that do not enjoy this property, syntactic inference does not match semantics [14]. The same paper introduces *stable rational closure* that solves the generality problem for rational closure, but re-introduces the issue of multiple (or non-existent) deductive closures. It is currently not clear how to design a logic that satisfies the KLM postulates, is fully general, and yields a unique closure for all knowledge bases.

Low complexity. \mathcal{DL}^N preserves the tractability of subsumption and instance checking for all low-complexity DLs, including the rich tractable logics \mathcal{EL}^{++} and DL-lite_{Horn}^(HN). Currently, no other nonmonotonic DL enjoys this property to the same extent. Rational closure has been proved to be tractable for \mathcal{EL} extended with \perp [19, 18]. Some logics, such as [13, 20, 21, 11, 10], preserve the asymptotic complexity of ExpTime-complete DLs like \mathcal{ALC} . More generally, \mathcal{DL}^N preserves the asymptotic complexity of all the DLs that belong to a deterministic complexity class that contains P. For nondeterministic complexity classes \mathcal{C} , an upper bound is $P^{\mathcal{C}}$.

In [15, 16] the semantic properties of \mathcal{DL}^N and the computational complexity of the related reasoning tasks have been thoroughly studied, here we focus on a concrete implementation of a \mathcal{DL}^N reasoner. As mentioned above, \mathcal{DL}^N preserves the tractability of low-complexity DLs; this opens the way to processing very large nonmonotonic KBs within these fragments. Asymptotic tractability alone, however, does not suffice for practical purposes. \mathcal{DL}^N reasoning is based on an iterative procedure that, given the signature of the queries of interest, discards overridden inclusions and transforms the other defeasible inclusions into classical axioms. This reduction executes a number of concept consistency checks that in the worst case is quadratic in the size of the knowledge base and of the input signature. Such consistency checks are evaluated against different, generally incomparable subsets of the knowledge base, and for this reason they cannot be implemented simply by computing a single classification of the knowledge base.

A preliminary implementation of a \mathcal{DL}^N reasoner in [15] has distinctly shown that such a quadratic dependence may slow down the computation significantly, even if the engine leverages the incremental reasoning facilities natively supported by state-of-art reasoners such as ELK. Consequently, practical reasoning on large knowledge bases – such as biomedical ontologies – requires ad hoc optimizations to detect and prune unnecessary computations during the reduction to classical DL. For this purpose, in this paper we introduce two optimization techniques, prove their correctness, and assess their effectiveness experimentally.

The first optimization aims at discarding axioms that are irrelevant for a given

query by adapting a classical module extraction algorithm to \mathcal{DL}^N . As a side effect, this reduces also the number of iterations in the reduction to classical DL. Adapting classical module extractors to \mathcal{DL}^N turns out to be a nontrivial task, given the nonmonotonic nature of its inferences.

As mentioned above, the first reasoner for \mathcal{DL}^N exploited the incremental reasoning mechanisms of the underlying classical reasoner. The second optimization, called *optimistic method*, aims at reducing the number of retractions (that are typically the most expensive operations in incremental reasoning).

We experimentally appraise the efficiency of each optimization and their combination. In order to obtain realistic test cases, experiments have been conducted on nonmonotonic versions of large biomedical ontologies such as Gene Ontology, Fly Anatomy, and SNOMED. Not only these ontologies are being used in concrete application scenarios, they are also commonly adopted in performance benchmarking [22, 23, 24, 25, 26, 27, 28]. The experimental results show that the two optimizations together reduce the computation time up to four orders of magnitude. A detailed analysis of their effectiveness is reported in Section 5.

We start with preliminaries on DL, module extraction, and \mathcal{DL}^N . Then, in Section 3, we introduce the two optimizations and prove their correctness and completeness. The implementation of the testbed and the test sets are described in Section 4. After the experimental evaluation (Section 5), two sections on related work and final remarks conclude the paper. Software and data can be downloaded from t.ly/Ct68.

2. Preliminaries

2.1. Description Logics

Description logics are a family of formal languages representing the logical foundations of the W3C Ontology Web Language (OWL2). DL languages are built from a set N_C of *concept names*, a set N_R of *role names*, and a set N_I of *individual names* (all countably infinite and pairwise disjoint). We use the term *predicate* to refer to elements of $N_C \cup N_R$. A signature Σ is a subset of $N_C \cup N_R \cup N_I$.

An interpretation \mathcal{I} of a signature Σ is a structure $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a nonempty set, and the interpretation function $\cdot^{\mathcal{I}}$, defined over Σ , is such that (i) $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ if $A \in N_C$; (ii) $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ if $R \in N_R$; (iii) $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ if $a \in N_I$. In each DL, compound concepts and roles are inductively defined with the help of a specific set of logical operators. Here, we focus on the DL *SR0IQ* which is the basis of the OWL2 standard. The constructors of *SR0IQ* are described in Table 1. We will use metavariables A, B for concept names, C, D for (possibly

compound) concepts, R, S for (possibly inverse) roles, a, b for individual names. The third column shows how to extend the valuation $\cdot^{\mathcal{I}}$ of an interpretation \mathcal{I} to compound expressions. Table 1 also shows the terminological and assertional axioms we deal with. An interpretation \mathcal{I} *satisfies* an axiom α (in symbols, $\mathcal{I} \models \alpha$) if it satisfies the corresponding semantic condition in Table 1. As usual, \perp , $\exists R.C \sqcup D$, and $\forall R.C$ are syntactic abbreviations for $\neg\top$, $\exists R.\top$, $\neg(\neg C \sqcap \neg D)$, and $\neg\exists R.\neg C$, respectively. Similarly, $C \equiv D$ is an abbreviation for the pair of inclusions $C \sqsubseteq D$ and $D \sqsubseteq C$.

A finite set of terminological axioms \mathcal{T} is called a TBox, similarly a finite set of assertional axioms \mathcal{A} is called an ABox. Then, a (classical) knowledge base is the union of a TBox and an ABox, $\mathcal{S} = \mathcal{T} \cup \mathcal{A}$.⁴

If X is a DL expression, an axiom, or a knowledge base, then $\text{sig}(X)$ denotes the signature consisting of all symbols occurring in X . An interpretation \mathcal{I} of a signature $\Sigma \supseteq \text{sig}(\mathcal{S})$ is a *model* of \mathcal{S} (in symbols, $\mathcal{I} \models \mathcal{S}$) if \mathcal{I} satisfies all the axioms in \mathcal{S} . We say that \mathcal{S} *entails* an axiom α (in symbols, $\mathcal{S} \models \alpha$) if all the models of \mathcal{S} satisfy α . Given a knowledge base \mathcal{S} and general concept inclusion (GCI) $C \sqsubseteq D$, the *subsumption problem* consists in deciding whether $\mathcal{S} \models C \sqsubseteq D$.

Hereafter, we denote by \mathcal{DL} a generic fragment of \mathcal{SROIQ} . We refer to [30, 31] for a comprehensive overview over DL fragments and for DL naming conventions.

2.2. Module extraction

Informally speaking, given a classical knowledge base \mathcal{S} and a reference signature Σ , a module is a subset \mathcal{M} of \mathcal{S} that preserves the consequences of \mathcal{S} that can be expressed with the symbols in Σ . Different notions of modules can be obtained by varying the class of such consequences (e.g. first-order, second-order, assertions, inclusions, etc.), see [32] for more details and possible applications.

Here, we use modules as an optimization technique for entailment checking over large ontologies: queries are evaluated against a module \mathcal{M} that is often much smaller than the knowledge base \mathcal{S} . Accordingly, we use the following definition of modules.

⁴Note that, in order to preserve decidability of the main reasoning tasks, a \mathcal{SROIQ} knowledge base must satisfy some additional conditions (e.g. regularity of RIA axioms and simplicity of the roles denoted in Table 1 by the symbol S). However, these conditions are not used in this paper, therefore we refer the reader to [29] for further details.

Name	Syntax	Semantics
Compound expressions		
inverse role	R^-	$\{(y, x) \mid (x, y) \in R^{\mathcal{I}}\} \quad (R \in \mathbf{N}_R)$
top	\top	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
universal role	\mathbf{U}	$\mathbf{U}^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
nominals	$\{a\}$	$\{a\}^{\mathcal{I}} = \{a^{\mathcal{I}}\}$
intersection	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
complement	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
existential restriction	$\exists R.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists (d, e) \in R^{\mathcal{I}} : e \in C^{\mathcal{I}}\}$
number restrictions	$(\bowtie n S.C)$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid (x, y) \in S^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \bowtie n\} \quad (\bowtie \doteq \leq, \geq)$
self	$\exists S.\text{Self}$	$\{x \in \Delta^{\mathcal{I}} \mid (x, x) \in S^{\mathcal{I}}\}$
Terminological axioms		\mathcal{I} satisfies the axiom if:
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
role disjointness	$\text{disj}(S_1, S_2)$	$S_1^{\mathcal{I}} \cap S_2^{\mathcal{I}} = \emptyset$
RIA	$R_1 \circ \dots \circ R_n \sqsubseteq R$	$R_1^{\mathcal{I}} \circ \dots \circ R_n^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
irreflexivity	$\text{irr}(S)$	$R^{\mathcal{I}}$ is irreflexive
Concept and role assertion axioms		
conc. assrt.	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assrt.	$R(a, b)$	$(a, b)^{\mathcal{I}} \in R^{\mathcal{I}}$

Table 1: Syntax and semantics of \mathcal{SROIQ} .

Definition 1. Let Σ be a signature and \mathcal{S} a DL knowledge base. A subset \mathcal{M} of \mathcal{S} is a Σ -module of \mathcal{S} iff, for all concept subsumptions α such that $\text{sig}(\alpha) \subseteq \Sigma$, it holds that $\mathcal{M} \models \alpha$ iff $\mathcal{S} \models \alpha$.

In general, deciding whether \mathcal{M} is a Σ -module of a knowledge base \mathcal{S} is an expensive reasoning task, even if \mathcal{S} does not contain assertions. For example, in the low-complexity description logics \mathcal{EL} and $DL\text{-}lite_{horn}$ the problem is EXPTIME-complete and coNP-complete, respectively [33, 34]; for \mathcal{ALC} , the asymptotic complexity is 2EXPTIME-complete [35], while for \mathcal{ALCQIO} the problem becomes even undecidable [36].

A more effective technique based on a notion of *locality* has been proposed to extract a Σ -module \mathcal{M} from \mathcal{S} [32, 37, 38]. The underlying idea is the following. A \perp -module of \mathcal{S} with respect to a given signature of interest Σ is a set $\mathcal{M} \subseteq \mathcal{S}$

such that all axioms $\alpha \in \mathcal{S} \setminus \mathcal{M}$ are \perp -local with respect to $\Sigma \cup \text{sig}(\mathcal{M})$. This means that if all concept names (resp. role names) that do not belong to $\Sigma \cup \text{sig}(\mathcal{M})$ are replaced with \perp (resp. the empty role), then α becomes a tautology. The definition of \top -modules and \top -locality are analogous, the only difference is that the concepts and the roles that are not in $\Sigma \cup \text{sig}(\mathcal{M})$ are replaced with \top and the universal role, respectively.

In order to make locality checking tractable, syntactic approximations of locality have been introduced [38]. This approach is implemented in the OWL API⁵ that support three module extractors. Two of them are denoted by $x\text{-Mod}(\Sigma, \mathcal{S})$, where $x \in \{\top, \perp\}$ indicates which notion of locality is used. The third is denoted by $\top\perp^*\text{-Mod}(\Sigma, \mathcal{S})$, and consists in an iterative alternation of the previous two, that is, $\top\perp^*\text{-Mod}(\Sigma, \mathcal{S})$ is the limit of the sequence:

$$\begin{aligned} \mathcal{M}_0 &= \mathcal{S}, \\ \mathcal{M}_{i+1} &= \top\text{-Mod}(\Sigma, \perp\text{-Mod}(\Sigma, \mathcal{M}_i)). \end{aligned}$$

Clearly $\top\perp^*\text{-Mod}(\Sigma, \mathcal{S})$ in general returns the smallest module among the three extractors.

Even if these techniques do not ensure that the extracted module is \subseteq -minimal, they turn out to be effective enough to be used as an optimization technique in many concrete cases.

We are going to use the three module extractors supported by the OWL API in our optimizations of nonmonotonic reasoning. Our correctness and completeness results, however, are slightly more general, as they apply to a class of module extractors of which the three extractors of the OWL API are particular instances. In the proofs contained in the following sections, it is convenient to reason with a declarative characterization of such class, as opposed to specific algorithmic characterizations. The declarative characterization is the following:

Definition 2. ($\top\perp$ -substitutions, locality, locality-based modules)

A $\top\perp$ -substitution for \mathcal{S} and a signature Σ is a substitution σ over $\text{sig}(\mathcal{S}) \setminus \Sigma$ that maps each concept name on \top or \perp , and each role name on the universal role or the empty role. An axiom α is σ -local iff $\sigma(\alpha)$ is a tautology. A set of axioms is σ -local if all of its members are. A set $\mathcal{M} \subseteq \mathcal{S}$ is a *locality-based Σ -module* of \mathcal{S} iff there exists a $\top\perp$ -substitution σ for \mathcal{S} and $\Sigma \cup \text{sig}(\mathcal{M})$ such that $\mathcal{S} \setminus \mathcal{M}$ is σ -local. In this case we say that σ *witnesses* that \mathcal{M} is a (locality-based) Σ -module of \mathcal{S} .

⁵See <https://github.com/owlcs/owlapi/wiki> for further details.

Let us remark again that each $x\text{-Mod}(\Sigma, \mathcal{S})$ ($x \in \{\perp, \top, \top\perp^*\}$) returns a locality-based Σ -module of \mathcal{S} .

All locality-based Σ -modules of any classical knowledge base \mathcal{S} are Σ -modules of \mathcal{S} (as per Definition 1). This result is implicit in [38, Prop. 42]; we include the proof of our version, for completeness:

Theorem 1. *For all classical knowledge bases \mathcal{S} and all signatures Σ , if \mathcal{M} is a locality-based Σ -module of \mathcal{S} , then \mathcal{M} is a Σ -module of \mathcal{S} in the sense of Definition 1.*

PROOF. Let \mathcal{M} be a locality-based Σ -module of \mathcal{S} , and let q be any subsumption query such that $\text{sig}(q) \subseteq \Sigma$. Let σ be a $\top\perp$ -substitution that witnesses that \mathcal{M} is a locality-based Σ -module of \mathcal{S} . We have to prove that $\mathcal{S} \models q$ iff $\mathcal{M} \models q$. The “if” part holds by the monotonicity of classical logic. We prove the “only if” part by reductio ad absurdum. Assume that $\mathcal{S} \models q$ but $\mathcal{M} \not\models q$. Then there exists a model \mathcal{I} of \mathcal{M} such that

$$\mathcal{I} \models \mathcal{M} \text{ and } \mathcal{I} \not\models q. \quad (1)$$

Define another interpretation \mathcal{J} as follows:

$$\begin{aligned} \Delta^{\mathcal{J}} &= \Delta^{\mathcal{I}}; \\ a^{\mathcal{J}} &= a^{\mathcal{I}} \quad \text{for all } a \in \mathbf{N}_I \\ X^{\mathcal{J}} &= \begin{cases} X^{\mathcal{I}} & \text{if } X \in \Sigma \cup \text{sig}(\mathcal{M}) \\ \sigma(X)^{\mathcal{I}} & \text{otherwise} \end{cases} \quad \text{for all } X \in \mathbf{N}_C \cup \mathbf{N}_R \end{aligned}$$

We are going to prove that

$$\mathcal{J} \models \mathcal{S} \text{ and } \mathcal{J} \not\models q, \quad (2)$$

which implies $\mathcal{S} \not\models q$, hence a contradiction (which completes the proof). It is easy to verify by structural induction that $Y^{\mathcal{J}} = \sigma(Y)^{\mathcal{I}}$, for any compound concept or role Y . Consequently, for all axioms γ we have that

- a) $\mathcal{J} \models \gamma$ iff $\mathcal{I} \models \sigma(\gamma)$;
- b) if $\text{sig}(\gamma) \subseteq \Sigma \cup \text{sig}(\mathcal{M})$, then $\mathcal{J} \models \gamma$ iff $\mathcal{I} \models \gamma$.

By b) and (1), it follows that

$$\mathcal{J} \models \mathcal{M} \text{ and } \mathcal{J} \not\models q. \quad (3)$$

Moreover, for all $\alpha \in \mathcal{S} \setminus \mathcal{M}$, α is σ -local, therefore $\sigma(\alpha)$ is a tautology. It follows by a) that $\mathcal{J} \models \mathcal{S} \setminus \mathcal{M}$. From this fact and (3) we conclude that (2) holds. \square

In general, module extractors are not correct under nonmonotonic semantics, because they are insensitive to the dependencies between predicates introduced by nonmonotonic inference. In Section 3.1, this issue will be illustrated in the context of \mathcal{DL}^N , see Example 3 and Example 4. An example for Circumscription can be found in [17, Example 6].

2.3. The logic \mathcal{DL}^N

Let \mathcal{DL} be any classical description logic language, and let \mathcal{DL}^N be the extension of \mathcal{DL} with a new concept name NC for each \mathcal{DL} concept C . The new concepts are called *normality concepts*.

A (canonical) \mathcal{DL}^N *knowledge base* is a disjoint union $\mathcal{KB} = \mathcal{S} \cup \mathcal{D}$ where \mathcal{S} is a finite set of \mathcal{DL} axioms (called *strong* or classical axioms) and \mathcal{D} is a finite set of *defeasible inclusions* (DIs, for short) that are expressions $C \sqsubseteq_n D$ where C is a \mathcal{DL} concept and D a \mathcal{DL}^N concept. If $\delta = (C \sqsubseteq_n D)$, then $\text{pre}(\delta)$ and $\text{con}(\delta)$ denote C and D , respectively. Informally speaking, the set of DIs satisfied by all the instances of a normality concept NC constitute the *prototype* associated to C .

DIs are prioritized by a strict partial order \prec . If $\delta_1 \prec \delta_2$, then δ_1 has higher priority than δ_2 . \mathcal{DL}^N solves automatically only the conflicts that can be settled using \prec ; any other conflict shall be resolved by the knowledge engineer (typically by adding suitable DIs). The expression $\mathcal{KB} \approx \alpha$ means that α is a \mathcal{DL}^N *consequence* of \mathcal{KB} . We do not report the model-theoretic definition of \approx and present only its reduction to classical reasoning [15], on which implementations are based. The classical reduction of \approx requires some preliminary definitions:

- For all DIs $\delta \in \mathcal{D}$ and all normality concepts NC , let

$$\delta^{NC} = (NC \sqcap \text{pre}(\delta) \sqsubseteq \text{con}(\delta)); \quad (4)$$

the informal meaning of δ^{NC} is: NC 's instances satisfy δ ;

- for all sets of \mathcal{DL} axioms \mathcal{S}' and all DIs δ , let $\mathcal{S}' \downarrow_{\prec\delta}$ denote the result of removing from \mathcal{S}' all the axioms δ_0^{NC} such that δ_0 's priority is not higher than δ 's:

$$\mathcal{S}' \downarrow_{\prec\delta} = \mathcal{S}' \setminus \{\delta_0^{NC} \mid \delta_0 \not\prec \delta\};$$

in other words, $\mathcal{S}' \downarrow_{\prec\delta}$ is the subset of \mathcal{S}' that may override δ ;

- finally, let $\delta_1, \dots, \delta_{|\mathcal{D}|}$ be an arbitrary *linearization* of (\mathcal{D}, \prec) , which means that $\{\delta_1, \dots, \delta_{|\mathcal{D}|}\} = \mathcal{D}$ and for all $i, j = 1, \dots, |\mathcal{D}|$, if $\delta_i \prec \delta_j$ then $i < j$.

The members of \mathcal{D} are processed in the order specified by the chosen linearization (it can be proved that the outcome is the same for all linearizations).

The classical translation of \mathcal{KB} , denoted by \mathcal{KB}^Σ , is a function of both \mathcal{KB} and a signature Σ that specifies which normality concepts occur in the queries of interest. In order to simplify notation in the proofs of Section 3, we allow Σ to contain also symbols that are not normality concepts, although the definition below is insensitive to such concepts. The classical translation \mathcal{KB}^Σ is defined by the following inductive construction, where $\Gamma = \Sigma \cup \text{sig}(\mathcal{KB})$ and $i = 1, 2, \dots, |\mathcal{D}|$:

$$\mathcal{KB}_0^\Sigma = \mathcal{S} \cup \{NC \sqsubseteq C \mid NC \in \Gamma\} \quad (5)$$

$$\mathcal{KB}_i^\Sigma = \mathcal{KB}_{i-1}^\Sigma \cup \{\delta_i^{NC} \mid NC \in \Gamma \text{ and } \mathcal{KB}_{i-1}^\Sigma \downarrow_{\prec \delta_i} \cup \{\delta_i^{NC}\} \not\models NC \sqsubseteq \perp\}. \quad (6)$$

In informal terms, the first step extends \mathcal{S} with axioms that state that the normal instances of a concept C are a fortiori instances of C . The construction proceeds by processing the DIs $\delta_i \in \mathcal{D}$ in decreasing priority order. If adding δ_i to the (higher priority) $\delta_j \prec \delta_i$ that have been previously selected for NC does not make NC inconsistent, as stated by (6), then δ_i^{NC} is included in \mathcal{KB}^Σ , otherwise δ_i^{NC} is discarded (i.e. overridden). The faithfulness of the above translation can be stated as follows:

Theorem 2 ([15]). *Let \mathcal{KB} be a \mathcal{DL}^N knowledge base, and let α be a subsumption or an assertion in \mathcal{DL}^N such that $\{NC \mid NC \in \text{sig}(\alpha)\} \subseteq \Sigma$. Then*

$$\mathcal{KB} \approx \alpha \text{ holds iff } \mathcal{KB}^\Sigma \models \alpha.$$

Note that \mathcal{DL}^N is entirely parametric w.r.t. the priority relation \prec . In [15, 17], for instance, the authors consider several priority relations, including the priority relation of rational closure:

$$\delta \prec \delta' \text{ iff } \text{rank}(\delta') > \text{rank}(\delta), \quad (7)$$

where $\text{rank}(\cdot)$ is the ranking function of rational closure [13, 21]. In this paper, for simplicity, we use in the examples a priority relation, called specificity, where the specific default properties of a concept C have higher priority than the more generic properties of the concepts that subsume C . The formal definition is the following (where \mathcal{S} denotes the strong axioms of the knowledge base and $C \sqsubseteq_S D$ means $\mathcal{S} \models C \sqsubseteq D$):

$$\delta \prec \delta' \text{ iff } \text{pre}(\delta) \sqsubseteq_S \text{pre}(\delta') \text{ and } \text{pre}(\delta') \not\sqsubseteq_S \text{pre}(\delta). \quad (8)$$

Our results, however, hold for all priority relations.

Concerning the computational complexity of \mathcal{DL}^N entailment, the following characterization holds.

Theorem 3 ([15]). *Let \mathcal{DL} be a DL fragment such that subsumption (resp. instance) checking in \mathcal{DL} belongs to a complexity class \mathcal{C} , and deciding the preference relation \prec belongs to $P^{\mathcal{C}}$.⁶ If \mathcal{DL} supports \sqcap in the left-hand side of inclusions, then subsumption (resp. instance) checking in \mathcal{DL}^N is in $P^{\mathcal{C}}$.*

It is well-known that, if $\mathcal{C} = \text{TIME}(f(n))$ and f grows at least as fast as a polynomial, then $P^{\mathcal{C}} = \mathcal{C}$. Then, as a corollary of Theorem 3, deciding whether an entailment $\mathcal{KB} \models \alpha$ holds is tractable in low complexity description logics such as $DL\text{-lite}_{horn}^{(HLN)}$ [39] and \mathcal{EL}^{++} [40]. Similarly, the logics with an EXP-complete classical subsumption problem, such as \mathcal{ALC} , preserve their complexity in \mathcal{DL}^N , while reasoning in \mathcal{SROIQ}^N is in $P^{N^2\text{ExpTime}}$.

We end this section with two examples. The first example is borrowed from [2] and shows a typical use of overriding in a biomedical taxonomy.

Example 1. Mammalian red blood cells are an exceptional class of eukariotic cells: the latter have a nucleus, while the former, in their mature stage, do not have a nucleus. By default, prototypical eukariotic cells have a nucleus, while prototypical mammalian red blood cells should not. This introduces two conflicting DIs with different priority. The encoding in \mathcal{ALC}^N is:

$$\text{EukCell} \sqsubseteq_n \exists \text{has_nucleus} \tag{9}$$

$$\text{MamRedBldCel} \sqsubseteq \text{EukCell} \tag{10}$$

$$\text{MamRedBldCel} \sqsubseteq_n \neg \exists \text{has_nucleus} . \tag{11}$$

By (10), specificity yields (11) \prec (9), that is, (11) has higher priority than (9). Accordingly, the enumeration of \mathcal{D} must be $\delta_1 = (11)$, $\delta_2 = (9)$. Let

$$\Gamma = \Sigma = \{\text{NEukCell}, \text{NMamRedBldCel}\} .$$

⁶ $P^{\mathcal{C}}$ is the class of all problems that can be solved by a deterministic Turing machine in polynomial time using an oracle for \mathcal{C} .

The translation \mathcal{KB}^Σ consists of (10) (the only axiom in \mathcal{S}) plus:

$$\text{NEukCell} \sqsubseteq \text{EukCell} \quad (12)$$

$$\text{NMamRedBldCel} \sqsubseteq \text{MamRedBldCel} \quad (13)$$

$$\text{NEukCell} \sqcap \text{MamRedBldCel} \sqsubseteq \neg \exists \text{has_nucleus} \quad (14)$$

$$\text{NMamRedBldCel} \sqcap \text{MamRedBldCel} \sqsubseteq \neg \exists \text{has_nucleus} \quad (15)$$

$$\text{NEukCell} \sqcap \text{EukCell} \sqsubseteq \exists \text{has_nucleus} . \quad (16)$$

where (12) and (13) are the axioms of the form $N_C \sqsubseteq C$ with $N_C \in \Gamma$, and (14) = $\delta_1^{\text{NEukCell}}$, (15) = $\delta_1^{\text{NMamRedBldCel}}$, and (16) = $\delta_2^{\text{NEukCell}}$. Note that \mathcal{KB}^Σ does not contain $\delta_2^{\text{NMamRedBldCel}}$ because, in NMamRedBldCel , δ_2 is overridden by δ_1 . By (12) and (13), inclusions (15) and (16) can be simplified to

$$\text{NMamRedBldCel} \sqsubseteq \neg \exists \text{has_nucleus} \quad (17)$$

$$\text{NEukCell} \sqsubseteq \exists \text{has_nucleus} \quad (18)$$

therefore normal eukariotic cells have a nucleus while normal mammalian red blood cells do not have a nucleus, as required. Moreover, by (17) and (18), we have $\mathcal{KB} \approx \text{NMamRedBldCel} \sqsubseteq \neg \text{NEukCell}$ which means that normal mammalian red blood cells are abnormal eukariotic cells. \square

The second example shows how, differently from other non-monotonic approaches as Circumscription, \mathcal{DL}^N does not silently neutralize unresolved conflicts between DIs, and hence it allows the ontology engineer to detect and fix them.

Example 2. (Semantic policy) Suppose that project coordinators are both administrative staff and research staff. By default, administrative staff are allowed to sign payments, while research staff are not. A conflict arises since both of these default policies apply to project coordinators. Formally, \mathcal{KB} can be formalized with:

$$\text{Admin} \sqsubseteq_n \exists \text{has_right.Sign} \quad (19)$$

$$\text{Research} \sqsubseteq_n \neg \exists \text{has_right.Sign} \quad (20)$$

$$\text{PrjCrd} \sqsubseteq \text{Admin} \sqcap \text{Research} \quad (21)$$

Leaving the conflict unresolved may cause a variety of security problems. If project coordinators should *not* sign payments, and the default policy is *open* (i.e. authorizations are granted by default), then failing to infer $\neg \exists \text{has_right.Sign}$ would improperly authorize the signing operation. Conversely, if the authorization is to be granted, then failing to prove $\exists \text{has_right.Sign}$ causes a *denial of*

service (that is, the user is unable to complete a legal operation). To prevent these problems, \mathcal{DL}^N makes the conflict visible by inferring $\mathcal{KB} \approx \text{NPrjCrd} \sqsubseteq \perp$ (showing that *PrjCrd*'s prototype is inconsistent). This can be proved by checking that $\mathcal{KB}^\Sigma \models \text{NPrjCrd} \sqsubseteq \perp$, where $\Sigma = \{\text{NPrjCrd}\}$. Then \mathcal{KB}^Σ consists of (21), $\text{NPrjCrd} \sqsubseteq \text{PrjCrd}$, and the translation of (19) *and* (20) (none overrides the other because none is more specific under any of the two priorities); such translation is:

$$\text{NPrjCrd} \sqcap \text{Admin} \sqsubseteq \exists \text{has_right.Sign}, \quad (22)$$

$$\text{NPrjCrd} \sqcap \text{Research} \sqsubseteq \neg \exists \text{has_right.Sign} \quad (23)$$

From the above axioms, the desired consequence $\text{NPrjCrd} \sqsubseteq \perp$ easily follows.

□

Further examples regarding biomedical and access-control domains can be found in [15]. A set of artificial examples, initially introduced in [41], is used in [17] to compare \mathcal{DL}^N with other non-monotonic approaches [41, 21, 6].

3. Optimizing the Computation of Overriding

This section introduces the two optimization techniques for the nonmonotonic description logic \mathcal{DL}^N , namely, module extraction and the optimistic evaluation strategy.

3.1. Module Extraction for \mathcal{DL}^N

In order to speed up reasoning, one might simply apply a module extractor for monotonic DLs to the classical translation \mathcal{KB}^Σ of \mathcal{KB} . However, the computation of \mathcal{KB}^Σ requires to solve one entailment problem for each inclusion δ_i^{NC} such that $\text{NC} \in \Sigma \cup \text{sig}(\mathcal{KB})$ and $\delta_i \in \mathcal{KB}$, as specified in (6). The set of all such δ_i^{NC} may grow quadratically with the size of \mathcal{KB} . Here we show how to avoid the processing of many of the above entailments by adapting module extraction so that it can be applied *before* translating \mathcal{KB} into a classical knowledge base. In this way, both the number of normality concepts NC and the number of DIs δ_i that are considered during the translation can be reduced. The following definition extends the notion of $\top \perp^*$ -module and locality to DIs.

Definition 3. (locality for DIs, pre-modules) Let σ range over $\top \perp$ -substitutions, and let \mathcal{KB} be a \mathcal{DL}^N knowledge base. A DI $C \sqsubseteq_n D$ is σ -local iff $C \sqsubseteq D$ is σ -local. A set $\mathcal{M} \subseteq \mathcal{KB}$ is a *pre-module* of \mathcal{KB} with respect to Σ iff \mathcal{M} is a locality-based Σ module of \mathcal{KB} (as per Definition 2, using the extended notion of locality to handle DIs, and replacing \mathcal{S} with \mathcal{KB}).

Remark 1. Pre-modules can be computed by adapting any of the available module extraction algorithms so as to parse DIs. A simple approach consists in encoding DIs as classical OWL2 inclusions and marking them with suitable annotations to distinguish them from strong inclusions. In this way, the existing module extractors can be applied as they are.

Note that the pre-modules of any *classical* \mathcal{KB} w.r.t. Σ are Σ -modules of \mathcal{KB} according to Theorem 1. When \mathcal{KB} contains some DIs, instead, the pre-modules of \mathcal{KB} are not modules, in general, because they do not preserve the consequences of \mathcal{KB} .

Example 3. Let $\Sigma = \{NA, B\}$ and $\mathcal{KB} = \{A \sqsubseteq B\}$. The $\top\perp$ -substitution σ for \mathcal{KB} and Σ such that $\sigma(A) = \perp$ witnesses that the empty set is a pre-module of \mathcal{KB} w.r.t. Σ . Let $\alpha = (NA \sqsubseteq B)$. Clearly $\emptyset \not\approx \alpha$ while $\mathcal{KB} \approx \alpha$, so the empty pre-module does not preserve the consequences of \mathcal{KB} whose symbols are in Σ . Obviously, the problem is that the valid axioms of the form $NC \sqsubseteq C$ are not explicitly included in \mathcal{KB} , therefore pre-module extraction does not detect the dependencies between NC and C . This particular problem could be solved by extracting the pre-module from the extension of \mathcal{KB} with these valid axioms, for a suitable set of relevant concepts NC . \square

Example 4. Let $\Sigma = \{NA, A, A_1, A_2, P\}$, and let $\mathcal{KB} = \{\delta_1, \delta_2\}$, where

$$\begin{aligned}\delta_1 &= A \sqsubseteq_n \forall P.NB; \\ \delta_2 &= A_1 \sqsubseteq_n A_2.\end{aligned}$$

The set $\mathcal{M} = \{\delta_2\}$ is a pre-module of \mathcal{KB} w.r.t. Σ . This is witnessed by a $\top\perp$ -substitution σ such that $\sigma(NB) = \top$. Let

$$\alpha = NA \sqsubseteq \forall P.(\neg A_1 \sqcup A_2).$$

It is not hard to see that $\mathcal{M} \not\approx \alpha$. On the contrary, $\mathcal{KB} \approx \alpha$ holds, therefore \mathcal{M} does not preserve the consequences of \mathcal{KB} . The \mathcal{DL}^N entailment $\mathcal{KB} \approx \alpha$ can be verified through the classical translation \mathcal{KB}^Σ , that contains the inclusions:

$$\begin{aligned}\delta_1^{NA} &= NA \sqcap A \sqsubseteq \forall P.NB, \text{ that due to } (NA \sqsubseteq A) \in \mathcal{KB}^\Sigma \text{ is equivalent to:} \\ &\quad NA \sqsubseteq \forall P.NB \\ \delta_2^{NB} &= NB \sqcap A_1 \sqsubseteq A_2 \equiv NB \sqsubseteq \neg A_1 \sqcup A_2.\end{aligned}$$

Thus, the inclusion α is a consequence of δ_1^{NA} and δ_2^{NB} . Let us analyze the source of the problem. Since δ_1 is not included in the module, $NB \notin \text{sig}(\mathcal{M})$. Then neither δ_1^{NA} nor δ_2^{NB} are considered in the translation \mathcal{M}^Σ . This time, the problem cannot be solved by extending \mathcal{KB} with $NA \sqsubseteq A$ and $NB \sqsubseteq B$. The substitution σ' such that $\sigma'(NB) = \sigma'(B) = \top$ witnesses that $\mathcal{M}' = \{\delta_2, NA \sqsubseteq A\}$ is a pre-module of $\mathcal{KB} \cup \{NA \sqsubseteq A, NB \sqsubseteq B\}$ w.r.t. Σ , still $\mathcal{M}' \not\approx \alpha$. The normality concepts occurring in \mathcal{KB} (like NB) should rather be protected by including them in the initial signature Σ . \square

The definition of N-modules below shows the correct way of extracting (an analogue of) a locality-based module from a \mathcal{DL}^N knowledge base \mathcal{KB} , under the reasonable assumption that the normality concepts of interest refer only to predicates that occur in \mathcal{KB} . More formally, hereafter we assume that:

Assumption 1. For all $NC \in \Sigma \cup \text{sig}(\mathcal{KB})$, $\text{sig}(C) \subseteq \text{sig}(\mathcal{KB})$.⁷

Definition 4. (N-modules) Let \mathcal{KB} be a \mathcal{DL}^N knowledge base, \mathcal{M} is an N-module of \mathcal{KB} with respect to a signature Σ iff \mathcal{M} is a pre-module of \mathcal{KB}^+ w.r.t. Γ where

1. $\Gamma = \Sigma \cup \{NC \mid NC \in \text{sig}(\mathcal{KB})\}$;
2. $\mathcal{KB}^+ = \mathcal{KB} \cup \{NC \sqsubseteq C \mid NC \in \Gamma\}$.

Note that \mathcal{M} is contained in \mathcal{KB}^+ and not in \mathcal{KB} , as the classical notion of modules would prescribe. \mathcal{M} is a subset of \mathcal{KB} extended with some \mathcal{DL}^N tautologies of the form $NC \sqsubseteq C$ that in general do not belong to \mathcal{KB} . However, we will verify experimentally that \mathcal{M} is nonetheless much smaller than \mathcal{KB} in many cases, so that the computation of \mathcal{M}^Σ is much faster than the computation of \mathcal{KB}^Σ , which is our final objective. Our next goal is proving that the additional elements in $\Gamma \setminus \Sigma$ and $\mathcal{KB}^+ \setminus \mathcal{KB}$ provide enough dependencies to ensure that $\mathcal{M} \approx \alpha$ iff $\mathcal{KB} \approx \alpha$, for all α such that $\text{sig}(\alpha) \subseteq \Sigma$. First, we need two technical lemmas on the properties of pre-modules. In particular, given a pre-module \mathcal{M} of \mathcal{KB} w.r.t. Σ , we show how to obtain from \mathcal{M} pre-modules for smaller \mathcal{KB} and Σ .

Lemma 4. Let \mathcal{M} be a pre-module of \mathcal{KB} w.r.t. a signature Σ and $\mathcal{KB}' \subseteq \mathcal{KB}$. Then $\mathcal{M} \cap \mathcal{KB}'$ is a pre-module of \mathcal{KB}' w.r.t. Σ .

⁷This assumption can always be satisfied by adding suitable tautologies to \mathcal{KB} .

PROOF. If \mathcal{M} is a pre-module of \mathcal{KB} w.r.t. Σ , then $\mathcal{KB} \setminus \mathcal{M}$ is σ -local for some $\top\perp$ -substitution σ for \mathcal{KB} and $\Sigma \cup \text{sig}(\mathcal{M})$. Let σ' be the restriction of σ to the symbols in

$$\text{sig}(\mathcal{KB}') \setminus (\Sigma \cup \text{sig}(\mathcal{M})) = \text{sig}(\mathcal{KB}') \setminus (\Sigma \cup \text{sig}(\mathcal{KB}' \cap \mathcal{M})).$$

Clearly, σ' is a $\top\perp$ -substitution for \mathcal{KB}' and $\Sigma \cup \text{sig}(\mathcal{M} \cap \mathcal{KB}')$. Moreover, for all $\beta \in \mathcal{KB}' \setminus \mathcal{M}$, we have by construction that $\sigma(\beta) = \sigma'(\beta)$. Since $\mathcal{KB}' \setminus \mathcal{M} = \mathcal{KB}' \setminus (\mathcal{M} \cap \mathcal{KB}')$, this implies that $\mathcal{KB}' \setminus (\mathcal{M} \cap \mathcal{KB}')$ is σ' -local and hence $\mathcal{M} \cap \mathcal{KB}'$ is a pre-module of \mathcal{KB}' w.r.t. Σ . \square

Lemma 5. *Let \mathcal{M} be a pre-module of \mathcal{KB} w.r.t. a signature Σ and let $\Sigma' \subseteq \Sigma$. Then, \mathcal{M} is a pre-module of \mathcal{KB} w.r.t. Σ' , too. Moreover, this can be witnessed by a function σ' such that if X is in the domain of σ' and $X \in \Sigma$, then $\sigma'(X) = \perp$.⁸*

PROOF. By hypothesis, there exists a $\top\perp$ -substitution σ for \mathcal{KB} and Σ , such that $\mathcal{KB} \setminus \mathcal{M}$ is σ -local w.r.t. $\Sigma \cup \text{sig}(\mathcal{M})$. Let σ' be the $\top\perp$ -substitution that extends σ from its domain $\text{dom} = \text{sig}(\mathcal{KB}) \setminus (\Sigma \cup \text{sig}(\mathcal{M}))$ to $\text{dom}' = \text{sig}(\mathcal{KB}) \setminus (\Sigma' \cup \text{sig}(\mathcal{M}))$ as follows:

$$\sigma'(X) = \begin{cases} \sigma(X) & \text{if } X \in \text{dom}; \\ \perp & \text{otherwise.} \end{cases}$$

Note that $\text{dom}' \setminus \text{dom} \subseteq \Sigma \setminus \Sigma'$. Clearly, σ' is a $\top\perp$ -substitution for \mathcal{KB} and $\Sigma' \cup \text{sig}(\mathcal{M})$. Now, consider an arbitrary $\alpha \in \mathcal{KB} \setminus \mathcal{M}$, and recall that all such α must be σ -local. By definition of locality, $\sigma(\alpha)$ is a tautology for all possible interpretations of the symbols in Σ . Consequently, α is still a tautology if the symbols in $\Sigma \cap \text{dom}'$ are replaced with \perp , as specified by σ' . This means that α is σ' -local. Then $\mathcal{KB} \setminus \mathcal{M}$ is σ' -local, and \mathcal{M} is a pre-module of \mathcal{KB} w.r.t. Σ' . Finally, by construction, we have that $\sigma'(X) = \perp$ for all $X \in \text{dom}' \cap \Sigma$. \square

Next we focus on N-modules. The proof that each N-module \mathcal{M} preserves all \mathcal{DL}^N -consequences is rather long; then, in order to improve readability, we outline the proof strategy before delving into the details. We rely on the faithfulness of the classical translations \mathcal{KB}^Σ and \mathcal{M}^Σ of \mathcal{KB} and \mathcal{M} (respectively); by proving that \mathcal{M}^Σ is a (classical) module of \mathcal{KB}^Σ , i.e.

$$\text{if } \text{sig}(\alpha) \subseteq \Sigma, \text{ then } \mathcal{M}^\Sigma \models \alpha \text{ iff } \mathcal{KB}^\Sigma \models \alpha, \quad (24)$$

⁸We replace all such X with \perp because we need this in the following lemmas. However the same proof holds for *all* the possible replacements of those X with any combination of \top and \perp .

\mathcal{KB}	The given knowledge base
Σ	The signature of the queries of interest
Γ	$\Sigma \cup \{NC \mid NC \in \text{sig}(\mathcal{KB})\}$
\mathcal{KB}^+	$\mathcal{KB} \cup \{NC \sqsubseteq C \mid NC \in \Gamma\}$
\mathcal{M}	An N-module of \mathcal{KB} w.r.t. Σ , that is, a pre-module of \mathcal{KB}^+ w.r.t. Γ .
$\Gamma_{\mathcal{M}}$	$\Sigma \cup \{NC \mid NC \in \text{sig}(\mathcal{M})\}$
\mathcal{M}_{all}	$\text{str}(\mathcal{M}) \cup \{\delta^{NC} \mid \delta \in \mathcal{M}, NC \in \Gamma_{\mathcal{M}}\}$
\mathcal{KB}_{all}	$\text{str}(\mathcal{KB}^+) \cup \{\delta^{NC} \mid \delta \in \mathcal{KB}, NC \in \Gamma\}$

Table 2: Summary of the symbols used in Section 3.1

we also show that if $\text{sig}(\alpha) \subseteq \Sigma$, then $\mathcal{M} \approx \alpha$ iff $\mathcal{KB} \approx \alpha$. Proving that \mathcal{M}^Σ is a module of \mathcal{KB}^Σ is not easy since the iterative construction of \mathcal{M}^Σ and \mathcal{KB}^Σ is nonmonotonic: any DI $\delta_i \in \mathcal{KB} \setminus \mathcal{M}$ might in principle override another DI δ_j , that is not overridden in \mathcal{M} . We obtain (24) by proving that

$$\mathcal{N} \subseteq \mathcal{M}^\Sigma \subseteq \mathcal{KB}^\Sigma$$

where \mathcal{N} is a suitable pre-module of \mathcal{KB}^Σ . This forces the three sets to have the same logical consequences α such that $\text{sig}(\alpha) \subseteq \Sigma$, which proves (24). The module \mathcal{N} is obtained by means of two upper approximations \mathcal{M}_{all} and \mathcal{KB}_{all} of \mathcal{M}^Σ and \mathcal{KB}^Σ , respectively. \mathcal{M}_{all} contains the strong part of \mathcal{M} plus all the inclusions δ_i^{NC} that are processed during the translation (including those that are eventually not included in \mathcal{M}^Σ). \mathcal{KB}_{all} has a similar relationship with \mathcal{KB}^+ . We first prove that \mathcal{M}_{all} is a pre-module of \mathcal{KB}_{all} . Then, by Lemma 4, $\mathcal{M}_{all} \cap \mathcal{KB}^\Sigma$ is a pre-module of \mathcal{KB}^Σ , and we set $\mathcal{N} = \mathcal{M}_{all} \cap \mathcal{KB}^\Sigma$.

The above proof steps will be formalized in reverse order. In the following, we need several auxiliary definitions and lemmas, where \mathcal{KB}^+ , Γ and \mathcal{M} are defined as in Definition 4 and $\text{str}(\mathcal{X})$ denotes the set of strong axioms in \mathcal{X} . Let:

1. $\Gamma_{\mathcal{M}} = \Sigma \cup \{NC \mid NC \in \text{sig}(\mathcal{M})\}$;
2. $\mathcal{M}_{all} = \text{str}(\mathcal{M}) \cup \{\delta^{NC} \mid \delta \in \mathcal{M}, NC \in \Gamma_{\mathcal{M}}\}$;
3. $\mathcal{KB}_{all} = \text{str}(\mathcal{KB}^+) \cup \{\delta^{NC} \mid \delta \in \mathcal{KB}, NC \in \Gamma\}$.

Table 2 summarizes the list of all symbols needed in the following.

Lemma 6. *\mathcal{M} is a pre-module of \mathcal{KB}^+ w.r.t. $\Gamma_{\mathcal{M}}$. This is witnessed by a $\top\perp$ -substitution σ' such that for all X in its domain, if $X \in \Gamma$ then $\sigma'(X) = \perp$.*

PROOF. We start by proving that $\Gamma_{\mathcal{M}} \subseteq \Gamma$. By definition $\Gamma_{\mathcal{M}} = \Sigma \cup \{NC \mid NC \in \text{sig}(\mathcal{M})\}$ and $\Gamma = \Sigma \cup \{NC \mid NC \in \text{sig}(\mathcal{KB})\}$, hence it suffices to show that $\{NC \mid NC \in \text{sig}(\mathcal{M})\} \subseteq \Gamma$.

Now recall that \mathcal{M} is a pre-module of \mathcal{KB}^+ w.r.t. Γ , by Definition 4. Then, \mathcal{M} consists by construction of two parts, $\mathcal{M}_1 \subseteq \mathcal{KB}$ and $\mathcal{M}_2 \subseteq \{NC \sqsubseteq C \mid NC \in \Gamma\}$. Let X be an arbitrary normality concept in $\text{sig}(\mathcal{M})$. If X occurs in \mathcal{M}_1 , we directly have that $X \in \Gamma$. Conversely, assume that X occurs in some inclusion $NC \sqsubseteq C$ in \mathcal{M}_2 ; if $X = NC$, then $X \in \Gamma$, otherwise $X \in \text{sig}(C)$ and, by Assumption 1, $\text{sig}(C) \subseteq \text{sig}(\mathcal{KB})$, therefore $X \in \Gamma$ holds also in this case.

Then, since $\Gamma_{\mathcal{M}} \subseteq \Gamma$ and \mathcal{M} is a pre-module of \mathcal{KB}^+ w.r.t. Γ , the Lemma follows directly from Lemma 5. \square

Lemma 7. \mathcal{M}_{all} is a pre-module of \mathcal{KB}_{all} w.r.t. $\Gamma_{\mathcal{M}}$.

PROOF. We have to show that there exists a $\top\perp$ -substitution σ for \mathcal{KB}_{all} and $\Gamma_{\mathcal{M}} \cup \text{sig}(\mathcal{M}_{all})$ such that $\mathcal{KB}_{all} \setminus \mathcal{M}_{all}$ is σ -local.

First, by Lemma 6 we know that there exists a $\top\perp$ -substitution σ' for \mathcal{KB}^+ and $\Gamma_{\mathcal{M}} \cup \text{sig}(\mathcal{M})$ such that $\mathcal{KB}^+ \setminus \mathcal{M}$ is σ' -local and such that for all X in the domain of σ' , if $X \in \Gamma$ then $\sigma'(X) = \perp$. By definition of $\top\perp$ -substitution, the domains of σ and σ' should be the following sets dom and dom' , respectively:

$$\text{dom} = \text{sig}(\mathcal{KB}_{all}) \setminus (\Gamma_{\mathcal{M}} \cup \text{sig}(\mathcal{M}_{all})); \quad (25)$$

$$\text{dom}' = \text{sig}(\mathcal{KB}^+) \setminus (\Gamma_{\mathcal{M}} \cup \text{sig}(\mathcal{M})). \quad (26)$$

Claim 1. $\text{dom}' \subseteq \text{dom}$.

To prove the claim, consider an arbitrary $X \in \text{dom}'$. The definition of \mathcal{KB}_{all} implies that $\text{sig}(\mathcal{KB}^+) \subseteq \text{sig}(\mathcal{KB}_{all})$; then, by (26), $X \in \text{sig}(\mathcal{KB}_{all})$. Moreover, (26) implies $X \notin \Gamma_{\mathcal{M}}$. Therefore, in order to prove that $X \in \text{dom}$, we only have to show that $X \notin \text{sig}(\mathcal{M}_{all})$. Suppose not. Then, by (26), X must occur in one of the axioms $\alpha \in \mathcal{M}_{all} \setminus \mathcal{M}$. By definition of \mathcal{M}_{all} , α must equal some δ^{NC} , where $\delta \in \mathcal{M}$ and $NC \in \Gamma_{\mathcal{M}}$. Note that $\text{sig}(\delta^{NC}) = \text{sig}(NC \sqcap \text{pre}(\delta) \sqsubseteq \text{con}(\delta)) = \text{sig}(\delta) \cup \{NC\}$. If $X = NC$, then $X \in \Gamma_{\mathcal{M}}$, hence $X \notin \text{dom}'$ (a contradiction). Then $X \in \text{sig}(\delta) \subseteq \text{sig}(\mathcal{M})$. But then $X \notin \text{dom}'$, and we reach a contradiction again. This completes the proof of Claim 1.

By Claim 1, we can define σ as an extension of σ' to dom . For all $X \in \text{dom}$, define:

$$\sigma(X) = \begin{cases} \sigma'(X) & \text{if } X \in \text{dom}' ; \\ \perp & \text{otherwise.} \end{cases}$$

As argued in the proof of Lemma 5, σ' -locality implies σ -locality, because σ extends σ' . With this property, we are going to prove that $\mathcal{KB}_{all} \setminus \mathcal{M}_{all}$ is σ -local (which completes the proof of this lemma).

Let α be an arbitrary member of $\mathcal{KB}_{all} \setminus \mathcal{M}_{all}$. By looking into the definitions of \mathcal{KB}_{all} and \mathcal{M}_{all} , it is easy to see that there are three possible cases:

- a) α is a strong inclusion in $\mathcal{KB}^+ \setminus \mathcal{M}$;
- b) $\alpha = \delta^{NC}$ and $\delta \in \mathcal{KB} \setminus \mathcal{M}$;
- c) $\alpha = \delta^{NC}$, $\delta \in \mathcal{M}$ and $NC \in \Gamma \setminus \Gamma_{\mathcal{M}}$.

In case a), α is σ' -local because $\mathcal{KB}^+ \setminus \mathcal{M}$ is σ' -local; then α is σ -local, too.

In case b), first note that $\delta \in \mathcal{KB}^+ \setminus \mathcal{M}$ (because $\mathcal{KB} \subseteq \mathcal{KB}^+$). Then, by analogy with case a), we derive that α is σ -local.

In case c), there are two possibilities. If $NC \in \text{dom} \setminus \text{dom}'$ then, by definition of σ , $\sigma(NC) = \perp$, so $\sigma(\delta^{NC}) = (\perp \sqcap \sigma(\text{pre}(\delta))) \sqsubseteq \sigma(\text{con}(\delta))$. This is clearly a tautology, therefore α is σ -local. Similarly, if $NC \in \text{dom}'$, $\sigma(NC) = \sigma'(NC) = \perp$ (because σ' maps all $X \in \Gamma$ on \perp), therefore $\sigma(\delta^{NC})$ is a tautology.

We conclude that in all cases α is σ -local, which completes the proof. \square

The next lemma makes use of the translation of \mathcal{M} into the classical knowledge base \mathcal{M}^Σ . In order to facilitate the comparison with \mathcal{KB}^Σ , the construction of \mathcal{M}^Σ is modified by inserting dummy steps for the DIs δ_i that are not included in \mathcal{M} . The modified translation of \mathcal{M} is the following (where \mathcal{D} is the set of DIs in \mathcal{KB} and $i > 0$):

$$\mathcal{H}_0^\Sigma = \text{str}(\mathcal{M}) \cup \{NC \sqsubseteq C \mid NC \in \Gamma_{\mathcal{M}}\} \quad (27)$$

$$\mathcal{H}_i^\Sigma = \begin{cases} \mathcal{H}_{i-1}^\Sigma & \text{if } \delta_i \notin \mathcal{M}; \text{ otherwise:} \\ \mathcal{H}_{i-1}^\Sigma \cup \{\delta_i^{NC} \mid NC \in \Gamma_{\mathcal{M}} \text{ and } \mathcal{H}_{i-1}^\Sigma \downarrow_{\prec \delta_i} \cup \{\delta_i^{NC}\} \not\sqsubseteq NC \sqsubseteq \perp\}. \end{cases} \quad (28)$$

By comparing the above translation with the translation \mathcal{M}^Σ defined by (5)–(6), it is easy to see that the only difference is that $\langle \mathcal{H}_i^\Sigma \rangle_i$ has an additional element $\mathcal{H}_i^\Sigma = \mathcal{H}_{i-1}^\Sigma$ for each $\delta_i \in \mathcal{KB} \setminus \mathcal{M}$, therefore the following proposition holds:

Proposition 8. $\mathcal{H}^\Sigma = \mathcal{M}^\Sigma$.

Lemma 9. *The following set containments hold:*

$$\mathcal{N} \subseteq \mathcal{M}^\Sigma \subseteq \mathcal{KB}^\Sigma,$$

where $\mathcal{N} = \mathcal{KB}^\Sigma \cap \mathcal{M}_{all}$.

PROOF. By Proposition 8, \mathcal{M}^Σ can be equivalent replaced with \mathcal{H}^Σ in the above chain of inclusions. Then, by definition of \mathcal{KB}^Σ and \mathcal{H}^Σ , it suffices to prove by induction that for all $i = 0, 1, \dots, |\mathcal{D}|$,

$$\mathcal{KB}_i^\Sigma \cap \mathcal{M}_{all} \subseteq \mathcal{H}_i^\Sigma \subseteq \mathcal{KB}_i^\Sigma.$$

Base case ($i = 0$): By definition of \mathcal{KB}_0^Σ and \mathcal{M}_{all} , $\mathcal{KB}_0^\Sigma \cap \mathcal{M}_{all}$ is equal to $\text{str}(\mathcal{M})$ which, in turn, is contained in \mathcal{H}_0^Σ . Moreover, by construction, $\mathcal{H}_0^\Sigma \subseteq \mathcal{KB}_0^\Sigma$.

Induction step ($i > 0$): The induction hypothesis (IH) is:

$$\mathcal{KB}_{i-1}^\Sigma \cap \mathcal{M}_{all} \subseteq \mathcal{H}_{i-1}^\Sigma \subseteq \mathcal{KB}_{i-1}^\Sigma.$$

First, suppose that $\delta_i \notin \mathcal{M}$ (and hence for all NC, $\delta_i^{\text{NC}} \notin \mathcal{M}_{all}$). Then $\mathcal{KB}_i^\Sigma \cap \mathcal{M}_{all} = \mathcal{KB}_{i-1}^\Sigma \cap \mathcal{M}_{all}$, $\mathcal{H}_i^\Sigma = \mathcal{H}_{i-1}^\Sigma$ (by construction), and $\mathcal{KB}_{i-1}^\Sigma \subseteq \mathcal{KB}_i^\Sigma$. The Lemma follows by IH.

Next assume that $\delta_i \in \mathcal{M}$ and let $\mathcal{F}_i = \{\delta_i^{\text{ND}} \mid \text{ND} \in \Gamma_{\mathcal{M}}\}$ and $\mathcal{G}_i = \{\delta_i^{\text{ND}} \mid \text{ND} \in \Gamma \setminus \Gamma_{\mathcal{M}}\}$. By construction,

$$\mathcal{KB}_i^\Sigma = \mathcal{KB}_{i-1}^\Sigma \cup (\mathcal{KB}_i^\Sigma \cap \mathcal{F}_i) \cup (\mathcal{KB}_i^\Sigma \cap \mathcal{G}_i). \quad (29)$$

Note that $\mathcal{G}_i \cap \mathcal{M}_{all} = \emptyset$ and $\mathcal{F}_i \subseteq \mathcal{M}_{all}$, by def. of \mathcal{M}_{all} . Consequently, we have that

$$\mathcal{KB}_i^\Sigma \cap \mathcal{M}_{all} = (\mathcal{KB}_{i-1}^\Sigma \cap \mathcal{M}_{all}) \cup (\mathcal{KB}_i^\Sigma \cap \mathcal{F}_i). \quad (30)$$

Moreover, since \mathcal{H}^Σ is constructed from $\Gamma_{\mathcal{M}}$, we also have that

$$\mathcal{H}_i^\Sigma = \mathcal{H}_{i-1}^\Sigma \cup (\mathcal{H}_i^\Sigma \cap \mathcal{F}_i). \quad (31)$$

By IH for each $\delta_i^{\text{ND}} \in \mathcal{F}_i$ it holds that

$$(\mathcal{KB}_{i-1}^\Sigma \cap \mathcal{M}_{all}) \downarrow_{\prec \delta_i} \cup \{\delta_i^{\text{ND}}\} \subseteq \mathcal{H}_{i-1}^\Sigma \downarrow_{\prec \delta_i} \cup \{\delta_i^{\text{ND}}\} \subseteq \mathcal{KB}_{i-1}^\Sigma \downarrow_{\prec \delta_i} \cup \{\delta_i^{\text{ND}}\}.$$

The leftmost term equals $(\mathcal{KB}_{i-1}^\Sigma \downarrow_{\prec \delta_i} \cup \{\delta_i^{\text{ND}}\}) \cap \mathcal{M}_{all}$. Since $(\mathcal{KB}_{i-1}^\Sigma \downarrow_{\prec \delta_i} \cup \{\delta_i^{\text{ND}}\}) \subseteq \mathcal{KB}_{all}$, by Lemmas 7 and 4, $(\mathcal{KB}_{i-1}^\Sigma \downarrow_{\prec \delta_i} \cup \{\delta_i^{\text{ND}}\}) \cap \mathcal{M}_{all}$ is a pre-module of $\mathcal{KB}_{i-1}^\Sigma \downarrow_{\prec \delta_i} \cup \{\delta_i^{\text{ND}}\}$ w.r.t. $\Gamma_{\mathcal{M}}$. Then, the leftmost term entails $\text{ND} \sqsubseteq \perp$ iff the rightmost does. It follows that the middle term $\mathcal{H}_{i-1}^\Sigma \downarrow_{\prec \delta_i} \cup \{\delta_i^{\text{ND}}\}$ entails $\text{ND} \sqsubseteq \perp$ iff the other two terms do. Then,

$$\mathcal{H}_i^\Sigma \cap \mathcal{F}_i = \mathcal{KB}_i^\Sigma \cap \mathcal{F}_i.$$

The Lemma follows from the above equation and (29)–(31) using the IH. \square

We are finally ready to prove that N-modules preserve \mathcal{DL}^N -reasoning (under Assumption 1):

Theorem 10. *Let \mathcal{M} be a N-module of \mathcal{KB} w.r.t. Σ . Then, for all α such that $\text{sig}(\alpha) \subseteq \Sigma$, $\mathcal{KB} \approx \alpha$ iff $\mathcal{M} \approx \alpha$.*

PROOF. By Lemma 7, \mathcal{M}_{all} is a pre-module of \mathcal{KB}_{all} w.r.t. $\Gamma_{\mathcal{M}} = \Sigma \cup \{NC \mid NC \in \text{sig}(\mathcal{M})\}$. Moreover, by definition, $\mathcal{KB}^\Sigma \subseteq \mathcal{KB}_{all}$. It follows, by Lemma 4, that $\mathcal{N} = \mathcal{KB}^\Sigma \cap \mathcal{M}_{all}$ is a pre-module of \mathcal{KB}^Σ w.r.t. $\Gamma_{\mathcal{M}}$.

Then, by Theorem 1, \mathcal{N} is also a classical $\Gamma_{\mathcal{M}}$ -module of \mathcal{KB}^Σ . Since $\Gamma_{\mathcal{M}} \supseteq \Sigma$, it follows that for all subsumptions α such that $\text{sig}(\alpha) \subseteq \Sigma$,

$$\mathcal{N} \models \alpha \text{ iff } \mathcal{KB}^\Sigma \models \alpha. \quad (32)$$

Then, by Lemma 9 we derive:

$$\mathcal{N} \subseteq \mathcal{M}^\Sigma \subseteq \mathcal{KB}^\Sigma.$$

From these inclusions, (32), and the monotonicity of \models , it follows that for all subsumptions α such that $\text{sig}(\alpha) \subseteq \Sigma$,

$$\mathcal{M}^\Sigma \models \alpha \text{ iff } \mathcal{KB}^\Sigma \models \alpha.$$

Equivalently, since the translation into classical DLs preserves \mathcal{DL}^N 's consequences, $\mathcal{M} \approx \alpha$ iff $\mathcal{KB} \approx \alpha$. \square

In our experiments we compute N-modules using the locality-based module $\top\perp^*$ -Mod of the OWL APIs, that is, we compute

$$\top\perp^*\text{-Mod}(\Gamma, \mathcal{KB}^+). \quad (33)$$

The extraction of N-modules can be profitably iterated. To see why, first note that (by Theorem 10) modules enjoy an obvious transitivity property:

Proposition 11. *If \mathcal{M} is a N-module of \mathcal{KB} w.r.t. Σ , and \mathcal{M}' is a N-module of \mathcal{M} w.r.t. Σ , then for all α such that $\text{sig}(\alpha) \subseteq \Sigma$, $\mathcal{KB} \approx \alpha$ iff $\mathcal{M}' \approx \alpha$.*

Classical module extractors are typically idempotent, i.e. the module \mathcal{M}' extracted from \mathcal{M} equals \mathcal{M} . On the contrary, N-module extraction implicitly includes in the initial signature Γ also the concepts NC that occur in \mathcal{KB} , some of which may not occur in \mathcal{M} ; thus, since \mathcal{M}' is constructed with the generally smaller initial

signature $\Gamma_{\mathcal{M}}$, it may consequently be a strict subset of \mathcal{M} . Thus, in Section 5, we analyze also the performance of a module extractor that iterates (33). In particular, the iterated module extractor computes the limit of the (monotonically decreasing) sequence:

$$\begin{aligned}\mathcal{M}_0 &= \mathcal{KB}^+ \\ \mathcal{M}_{i+1} &= \top\perp^*\text{-Mod}(\Gamma_i, \mathcal{KB}_i^+),\end{aligned}$$

where $\Gamma_i = \Sigma \cup \{\text{NC} \mid \text{NC} \in \text{sig}(\mathcal{M}_i)\}$ and $\mathcal{KB}_i^+ = \mathcal{M}_i \cup \{\text{NC} \sqsubseteq C \mid \text{NC} \in \Gamma_i\}$. It can be seen that the translation's size reduction can be quadratic, in the best case. The experiments in Section 5 confirm the effectiveness of the iterated N-module extractor.

3.2. Optimistic Computation

The construction of \mathcal{KB}^Σ repeats the concept consistency check (6) over knowledge bases of the form $\mathcal{KB}_{i-1}^\Sigma \downarrow_{\prec\delta_i} \cup \{\delta_i^{\text{NC}}\}$ that share a (possibly large) common part \mathcal{KB}_0^Σ , so incremental reasoning mechanisms tend to improve efficiency by avoiding multiple computations of the consequences of \mathcal{KB}_0^Σ . On the contrary, the set of inclusions of the form δ_j^{NC} that belong to $\mathcal{KB}_{i-1}^\Sigma \downarrow_{\prec\delta_i}$ may change significantly at each step due to the filtering $\downarrow_{\prec\delta_i}$. This operation requires many axiom deletions, which are less efficient than monotonically increasing changes. The optimistic algorithm introduced here (Algorithm 1) computes a knowledge base \mathcal{KB}^* equivalent to \mathcal{KB}^Σ in a way that tends to reduce the number of deletions.

Phase 1 of Algorithm 1 optimistically assumes that the DIs incomparable with δ_i^{NC} do not contribute to entailing $\text{NC} \sqsubseteq \perp$ in (6), so they are not filtered with $\downarrow_{\prec\delta_i}$ in line 7. In this way, the knowledge base used in line 7 increases monotonically during the for-do loop (incremental reasoning techniques are particularly efficient in this case). Phase 2 checks whether the DIs discarded during Phase 1 should actually be overridden; it does so by applying $\downarrow_{\prec\delta_j}$ in line 17. If the test in line 17 is true then it means that δ_j^{ND} is *not* overridden. However, since the corresponding test in line 7 (with $\delta_i^{\text{NC}} = \delta_j^{\text{ND}}$) was false, then there must be an unresolved conflict between δ_j^{ND} and some other DI with equal or incomparable priority, that makes ND inconsistent. Instead of including in the translation of \mathcal{KB} the translated DIs that entail the inconsistency of ND, Algorithm 1 directly asserts $\text{ND} \sqsubseteq \perp$ (line 18), which is logically equivalent.

Rather than carrying out Phase 2 by scanning the list of discarded DIs from beginning to the end, Algorithm 1 visits the list along chains in decreasing priority order (cf. the condition in the while-statement at line 15, and the assignment at

Algorithm 1: Optimistic-Method

Input: $\mathcal{KB} = \mathcal{S} \cup \mathcal{D}, \Sigma$
Output: a knowledge base \mathcal{KB}^* such that $\mathcal{KB}^* \equiv \mathcal{KB}^\Sigma$

// Phase 1

- 1 compute a linearization $\delta_1, \dots, \delta_{|\mathcal{D}|}$ of \mathcal{D}
- 2 $\Pi := \emptyset$ // Π collects the prototypes
- 3 $\Delta := \emptyset$ // ordered list of all discarded δ_i^{NC}
- 4 **for** $i = 1, 2, \dots, |\mathcal{D}|$ **do**
- 5 **for** $\text{NC} \in \Sigma$ **do**
- 6 $\Pi' := \Pi \cup \{\delta_i^{\text{NC}}\}$
- 7 **if** $\mathcal{KB}_0^\Sigma \cup \Pi' \not\models \text{NC} \sqsubseteq \perp$ **then**
- 8 $\Pi := \Pi'$
- 9 **else**
- 10 append δ_i^{NC} to Δ

// Phase 2

- 11 $\mathcal{KB}^* = \mathcal{KB}_0^\Sigma \cup \Pi$
- 12 **while** $\Delta \neq \emptyset$ **do**
- 13 read from Δ its first element δ_i^{NC} without removing it
- 14 let $\delta := \delta_i$
- 15 **while** Δ contains some δ_j^{ND} such that $\delta = \delta_j$ or $\delta \prec \delta_j$ **do**
- 16 remove from Δ the first such δ_j^{ND} // (in decreasing priority order)
- 17 **if** $(\mathcal{KB}_0^\Sigma \cup \Pi) \downarrow_{\prec \delta_j} \cup \{\delta_j^{\text{ND}}\} \not\models \text{ND} \sqsubseteq \perp$ **then**
- 18 $\mathcal{KB}^* := \mathcal{KB}^* \cup \{\text{ND} \sqsubseteq \perp\}$
- 19 extract all δ_k^{NE} with $E = D$ from Δ
- 20 let $\delta := \delta_j$

line 20), because in this way the knowledge bases used in line 17 often increase monotonically at each iteration despite the filtering operation $\downarrow_{\prec\delta_j}$ (so that incremental reasoning techniques are used efficiently also in Phase 2).

Example 5. In this first example we show how the optimistic algorithm may reduce the number of assertions and retractions. Let the knowledge base \mathcal{KB} contain the following axioms:

$$\begin{aligned} A_1 &\sqsubseteq A_2 \\ B_1 &\sqsubseteq B_2 \\ \delta_1 : A_1 &\sqsubseteq_n A \\ \delta_2 : B_1 &\sqsubseteq_n B \\ \delta_3 : A_2 &\sqsubseteq_n B \\ \delta_4 : B_2 &\sqsubseteq_n \neg B \end{aligned}$$

The important properties here are that, by specificity,

$$\begin{aligned} \delta_1 &\prec \delta_3 \\ \delta_2 &\prec \delta_4 \end{aligned}$$

and that δ_2 overrides δ_4 , while δ_1 and δ_3 are not in conflict with each other.

Suppose that the reasoning algorithms use the following linearization of the priority order:

$$\delta_1, \delta_2, \delta_3, \delta_4,$$

and suppose for a start that the queries of interest concern NB_1 , therefore $\Sigma = \{NB_1\}$.

The naive construction defined by (5) and (6) makes inferences with the following sequence of knowledge bases:

$$\mathcal{KB}_0^\Sigma \downarrow_{\prec\delta_1} \cup \{\delta_1^{NB_1}\} = \mathcal{KB}_0^\Sigma \cup \{\delta_1^{NB_1}\} \quad (34)$$

$$\mathcal{KB}_1^\Sigma \downarrow_{\prec\delta_2} \cup \{\delta_2^{NB_1}\} = \mathcal{KB}_0^\Sigma \cup \{\delta_2^{NB_1}\} \quad (35)$$

$$\mathcal{KB}_2^\Sigma \downarrow_{\prec\delta_3} \cup \{\delta_3^{NB_1}\} = \mathcal{KB}_0^\Sigma \cup \{\delta_1^{NB_1}\} \cup \{\delta_3^{NB_1}\} \quad (36)$$

$$\mathcal{KB}_3^\Sigma \downarrow_{\prec\delta_4} \cup \{\delta_4^{NB_1}\} = \mathcal{KB}_0^\Sigma \cup \{\delta_2^{NB_1}\} \cup \{\delta_4^{NB_1}\}. \quad (37)$$

Thus, in order to compute (34), $\delta_1^{NB_1}$ must be added (asserted) to \mathcal{KB}_0^Σ ; then in order to compute (35), $\delta_1^{NB_1}$ must be retracted and replaced with $\delta_2^{NB_1}$; (36) is

obtained by retracting $\delta_2^{NB_1}$ and asserting $\delta_1^{NB_1}$ and $\delta_3^{NB_1}$; (37) is obtained by retracting $\delta_1^{NB_1}$ and $\delta_3^{NB_1}$, and asserting $\delta_2^{NB_1}$ and $\delta_4^{NB_1}$. The totals are: 6 assertions and 4 retractions.

The optimistic algorithm, in the four iterations of the for-loop at line 4, progressively asserts $\delta_1^{NB_1}$, $\delta_2^{NB_1}$, $\delta_3^{NB_1}$, and $\delta_4^{NB_1}$. The first three inclusions are inserted in Π , while $\delta_4^{NB_1}$ is inserted in Δ . Then the algorithm proceeds to the second phase. It extracts $\delta_4^{NB_1}$ from Δ and retracts $\delta_1^{NB_1}$ and $\delta_3^{NB_1}$ to compute the knowledge base at line 17, that is:

$$(\mathcal{KB}_0^\Sigma \cup \Pi) \downarrow_{\prec \delta_4} \cup \{\delta_4^{NB_1}\} = \mathcal{KB}_0^\Sigma \cup \{\delta_2^{NB_1}, \delta_4^{NB_1}\}.$$

The totals are: 4 assertions and 2 retractions. In this case, the optimistic method saves 33% assertions and 50% retractions. The benefits of this processing strategy become more evident as the number of parallel chains of DIs grows. \square

Example 6. In this example we illustrate the benefits of processing Δ along chains of DIs with non-increasing priority (cf. the condition in the while-statement at line 15, and the assignment at line 20). We compare the number of assertions and retractions performed by phase 2 with those that would be executed if the list Δ were processed sequentially.

Suppose that the knowledge base contains five DIs $\delta_1, \dots, \delta_5$ such that

$$\delta_1 \prec \delta_3 \prec \delta_5 \text{ and } \delta_2 \prec \delta_4,$$

and that δ_1 overrides δ_3 and δ_5 while δ_2 overrides δ_4 . Here is a knowledge base with these properties:

$$\begin{aligned} A_0 &\sqsubseteq A_1 \sqcap B_1 \\ A_1 &\sqsubseteq A_2 \\ A_2 &\sqsubseteq A_3 \\ B_1 &\sqsubseteq B_2 \\ \delta_1 : & A_1 \sqsubseteq A' \sqcap A'' \\ \delta_3 : & A_2 \sqsubseteq \neg A' \\ \delta_5 : & A_3 \sqsubseteq \neg A'' \\ \delta_2 : & B_1 \sqsubseteq B' \\ \delta_4 : & B_2 \sqsubseteq \neg B' \end{aligned}$$

Let $\Sigma = \{NA_0\}$ and the linearization of the priority order be $\delta_1, \delta_2, \delta_3, \delta_4, \delta_5$.

At the end of the first phase of the optimistic algorithm, $\Pi = \{\delta_1^{NA_0}, \delta_2^{NA_0}\}$ and $\Delta = \langle \delta_3^{NA_0}, \delta_4^{NA_0}, \delta_5^{NA_0} \rangle$. If Δ were processed left-to-right, then the sequence of knowledge bases used in the test at line 17 would be:

$$(\mathcal{KB}_0^\Sigma \cup \Pi) \downarrow_{\neg \delta_3} \cup \{\delta_3^{NA_0}\} = \mathcal{KB}_0^\Sigma \cup \{\delta_1^{NA_0}, \delta_3^{NA_0}\} \quad (38)$$

$$(\mathcal{KB}_0^\Sigma \cup \Pi) \downarrow_{\neg \delta_4} \cup \{\delta_4^{NA_0}\} = \mathcal{KB}_0^\Sigma \cup \{\delta_2^{NA_0}, \delta_4^{NA_0}\} \quad (39)$$

$$(\mathcal{KB}_0^\Sigma \cup \Pi) \downarrow_{\neg \delta_5} \cup \{\delta_5^{NA_0}\} = \mathcal{KB}_0^\Sigma \cup \{\delta_1^{NA_0}, \delta_5^{NA_0}\}. \quad (40)$$

The knowledge base (38) is obtained from $\mathcal{KB}_0^\Sigma \cup \Pi$ by retracting $\delta_2^{NA_0}$ and asserting $\delta_3^{NA_0}$; then (39) is obtained from (38) by retracting $\delta_1^{NA_0}$ and $\delta_3^{NA_0}$, and by asserting $\delta_2^{NA_0}$ and $\delta_4^{NA_0}$; finally, (40) is obtained by retracting $\delta_2^{NA_0}$ and $\delta_4^{NA_0}$ and asserting $\delta_1^{NA_0}$ and $\delta_5^{NA_0}$. With this approach, before processing δ_5 , $\delta_1^{NA_0}$ is re-asserted and $\delta_2^{NA_0}$ is retracted again.

The second phase of the optimistic algorithm, instead, processes Δ in the following order: $\delta_3^{NA_0}, \delta_5^{NA_0}, \delta_4^{NA_0}$. The sequence of knowledge bases used at line 17 is the following:

$$(\mathcal{KB}_0^\Sigma \cup \Pi) \downarrow_{\neg \delta_3} \cup \{\delta_3^{NA_0}\} = \mathcal{KB}_0^\Sigma \cup \{\delta_1^{NA_0}, \delta_3^{NA_0}\} \quad (41)$$

$$(\mathcal{KB}_0^\Sigma \cup \Pi) \downarrow_{\neg \delta_5} \cup \{\delta_5^{NA_0}\} = \mathcal{KB}_0^\Sigma \cup \{\delta_1^{NA_0}, \delta_5^{NA_0}\} \quad (42)$$

$$(\mathcal{KB}_0^\Sigma \cup \Pi) \downarrow_{\neg \delta_4} \cup \{\delta_4^{NA_0}\} = \mathcal{KB}_0^\Sigma \cup \{\delta_2^{NA_0}, \delta_4^{NA_0}\}. \quad (43)$$

In this way, when $\delta_5^{NA_0}$ is processed, $\delta_1^{NA_0}$ needs not be re-asserted, and $\delta_2^{NA_0}$ needs not be retracted. The benefits of this processing strategy become more evident as the number of parallel chains of DIs grows. \square

In the above examples, all the overridings hypothesized in the first phase are confirmed in the second phase. The next example, instead, illustrates a case in which an unresolved conflict is detected in the second phase.

Example 7. Consider a slight modification of the previous example, where δ_1 and δ_3 have the same priority (e.g. replace δ_3 with $A_1 \sqsubseteq \neg A'$). At the end of the first phase, Π and Δ are the same as in Example 6. The first time line 17 is executed (with $j = 3$ and $D = A_0$), the test succeeds, so $NA_0 \sqsubseteq \perp$ is included in the translation, Δ is emptied, and the optimistic algorithm terminates. As a consequence, $\delta_4^{NA_0}$ and $\delta_5^{NA_0}$ need not be processed. In terms of assertions and retractions, it is easy to see that the second phase only needs to retract $\delta_2^{NA_0}$ from $\mathcal{KB}_0^\Sigma \cup \Pi$ and assert $\delta_3^{NA_0}$. \square

The following theorem shows the correctness of Algorithm 1 under the assumption that normality concepts do not occur in \mathcal{KB} (they may occur in the queries, though). We call such knowledge bases *N-free*.⁹

Theorem 12. *If \mathcal{KB} is N-free, then Algorithm 1's output is equivalent to \mathcal{KB}^Σ .*

PROOF. First assume that Σ is a singleton ($|\Sigma| = 1$) and let NC be its unique member. We start by proving some invariants of lines 6-10.

Claim 1: $\mathcal{KB}^\Sigma \models \Pi$.

Claim 2: If, for some $j < i$, $\delta_j^{NC} \in \mathcal{KB}^\Sigma \setminus \Pi$, then $\mathcal{KB}^\Sigma \models NC \sqsubseteq \perp$.

We prove these two claims simultaneously by induction on the iteration i . Both claims hold vacuously at the first execution of line 6. Next, assume by induction hypothesis that they hold at line 6 in some iteration; we have to prove that they still hold at the next iteration. There are two possibilities: First suppose that for some $j < i$, $\delta_j^{NC} \in \mathcal{KB}^\Sigma \setminus \Pi$. By Claim 2, $\mathcal{KB}^\Sigma \models NC \sqsubseteq \perp$. This immediately implies that Claim 2 holds also at the next iteration. Moreover, it implies Claim 1 because all members of Π have an occurrence of NC in the left-hand side.

We are left the case in which

$$\text{for all } j < i, \text{ if } \delta_j^{NC} \in \mathcal{KB}^\Sigma \text{ then } \delta_j^{NC} \in \Pi. \quad (44)$$

If the condition in line 7 is false, then Π is not changed, so Claim 1 must hold at the next iteration. Otherwise, by (44),

$$\mathcal{KB}_0^\Sigma \cup \Pi' \supseteq \mathcal{KB}_{i-1}^\Sigma \downarrow_{\delta_i} \cup \{\delta_i^{NC}\}, \quad (45)$$

and hence $NC \sqsubseteq \perp$ is not provable in (6), either. It follows that δ_i^{NC} belongs to both Π (by line 8) and \mathcal{KB}^Σ (by (6)). This proves Claim 1 for iteration i .

Concerning Claim 2, first suppose that the condition in line 7 is false; then either Claim 2 remains vacuously satisfied, or $\delta_i^{NC} \in \mathcal{KB}^\Sigma \setminus \Pi$. The latter (plus the definition of \mathcal{KB}^Σ and the IH of Claim 1) implies that $\mathcal{KB}_0^\Sigma \cup \Pi'$ is entailed by \mathcal{KB}^Σ . It follows that $\mathcal{KB}^\Sigma \models NC \sqsubseteq \perp$ as well, which proves Claim 2 in this case. Finally, if the condition in line 7 is true, then at line 8 $\delta_i^{NC} \in \Pi$. Together with (44), this implies that Claim 2 holds vacuously.

Claim 3: If $\mathcal{KB}^\Sigma \models NC \sqsubseteq \perp$ then $\mathcal{KB}^* \models NC \sqsubseteq \perp$.

Suppose not (we shall derive a contradiction). The assumption and Claim 1 imply that there must be some $\delta_k^{NC} \in \mathcal{KB}^\Sigma \setminus \mathcal{KB}^*$. Let δ_i^{NC} be the one with

⁹Note that the examples in Section 2.3 are all N-free.

minimal k . Using minimality, it can be proved that $\mathcal{KB}_0^\Sigma \cup \Pi \downarrow_{\prec \delta_i} = \mathcal{KB}_{i-1}^\Sigma \downarrow_{\prec \delta_i}$, so the concept consistency test in line 17 (the latter instantiated with $j = i$ and $D = C$) is equivalent to the one in (6). But then $\delta_i^{NC} \in \mathcal{KB}^\Sigma$ iff $\delta_i^{NC} \in \mathcal{KB}^*$, which contradicts the assumption, so Claim 3 is proved.

Claim 4: If $\mathcal{KB}^\Sigma \models NC \sqsubseteq \perp$ then $\mathcal{KB}^* \equiv \mathcal{KB}^\Sigma$.

Note that $\mathcal{KB}^* \subseteq \mathcal{KB}_0^\Sigma \cup \Pi \cup \{NC \sqsubseteq \perp\}$ (cf. lines 11, and 18). Clearly, $\mathcal{KB}^\Sigma \models \mathcal{KB}_0^\Sigma \cup \Pi \cup \{NC \sqsubseteq \perp\}$ (by def., Claim 1 and the assumption), so $\mathcal{KB}^\Sigma \models \mathcal{KB}^*$. We are left to prove $\mathcal{KB}^* \models \mathcal{KB}^\Sigma$. By Claim 3, $\mathcal{KB}^* \models NC \sqsubseteq \perp$, and this inclusion in turn entails all $\delta_i^{NC} \in \mathcal{KB}^\Sigma$ (cf. (4)). The other members of \mathcal{KB}^Σ are those in \mathcal{KB}_0^Σ , by definition, and $\mathcal{KB}^* \supseteq \mathcal{KB}_0^\Sigma$ (line 11). It follows that $\mathcal{KB}^* \models \mathcal{KB}^\Sigma$, which completes the proof of Claim 4.

Claim 5: If $\mathcal{KB}^\Sigma \not\models NC \sqsubseteq \perp$ then $\mathcal{KB}^* \equiv \mathcal{KB}^\Sigma$.

Suppose that $\mathcal{KB}^\Sigma \not\models NC \sqsubseteq \perp$. Then, by the contrapositive of Claim 2 all δ^{NC} are in Π . Hence, it follows by Claim 1 that $\mathcal{KB}_0^\Sigma \cup \Pi \equiv \mathcal{KB}^\Sigma$ which further implies that the concept consistency in line 17 is equivalent to the corresponding test in (6). Then it can be proved that if any of these tests were true, then also $\mathcal{KB}^\Sigma \models NC \sqsubseteq \perp$ because, by Π 's construction, in that case δ_i must be in conflict with some other DI with the same priority. However, $\mathcal{KB}^\Sigma \not\models NC \sqsubseteq \perp$ contradicts the assumption. It follows that all tests in line 17 are false, so $\mathcal{KB}^* = \mathcal{KB}_0^\Sigma \cup \Pi$, and we have already argued that this knowledge base is equivalent to \mathcal{KB}^Σ . This completes the proof for $|\Sigma| = 1$.

For $|\Sigma| > 1$, note that the tests in lines 7 and 17 do not depend on any δ_k^{NE} such that $E \neq C$ (resp. $E \neq D$). Indeed, by \perp -locality, all such δ_k^{NE} are local w.r.t. the signature of $\mathcal{KB} \cup \{NC \sqsubseteq \perp\}$, so they can be removed without changing the result of the concept consistency test [32]. However, after their removal, the concept consistency tests correspond to the ones for the singleton case $\Sigma = \{NC\}$, which we have already proved correct. \square

We conclude this section by showing that when \mathcal{KB} is not N-free, the output of Algorithm 1 is not always correct:

Example 8. Let \mathcal{KB} consists of the following axioms (where δ_3 is not N-free):

$$\begin{aligned} & A \sqsubseteq B \\ \delta_1 : & A \sqsubseteq_n C \\ \delta_2 : & A \sqsubseteq_n \neg C \\ \delta_3 : & B \sqsubseteq_n \exists R.NA. \end{aligned}$$

Specificity gives δ_1 and δ_2 the same priority, and, by the first axiom above, δ_3 has lower priority than the other two DIs. In order to answer queries about NA and

NB , let $\Sigma = \{NA, NB\}$. It is not hard to see that the two conflicting DIs with the same priority (i.e. δ_1 and δ_2) make NA inconsistent; in particular, \mathcal{KB}^Σ contains both

$$\begin{aligned}\delta_1^{NA} &= NA \sqcap A \sqsubseteq C \quad \text{and} \\ \delta_2^{NA} &= NA \sqcap A \sqsubseteq \neg C\end{aligned}$$

that together with the axiom $(NA \sqsubseteq A) \in \mathcal{KB}^\Sigma$ imply $NA \sqsubseteq \perp$. As a consequence of this fact, δ_3 is overridden in NB ; in particular, the inclusion

$$\delta_3^{NB} = NB \sqcap B \sqsubseteq \exists R.NA$$

is *not* included in \mathcal{KB}^Σ , because (together with $(NB \sqsubseteq B) \in \mathcal{KB}^\Sigma$ and $NA \sqsubseteq \perp$, that is implied by higher priority axioms) it implies $NB \sqsubseteq \perp$. Since δ_3 is overridden, NB is consistent ($\mathcal{KB}^\Sigma \not\models NB \sqsubseteq \perp$). On the contrary, Algorithm 1 returns a knowledge base \mathcal{KB}^* that erroneously entails $NB \sqsubseteq \perp$. This happens as follows, assuming that the linearization is $\delta_1, \delta_2, \delta_3$:¹⁰ During Phase 1, the translations of the above DIs are processed in the following order:

- first δ_1^{NA} and δ_1^{NB} , in some order; none of them makes the respective normality concept inconsistent, so they are both inserted in Π ;
- then δ_2^{NA} and δ_2^{NB} ; the former, together with δ_1^{NA} and \mathcal{KB}_0^Σ , implies $NA \sqsubseteq \perp$, so it is inserted in Δ ;
- finally δ_3^{NA} and δ_3^{NB} are processed. In the absence of δ_2^{NA} (that has been temporarily discarded), NA is consistent, therefore δ_3^{NA} and δ_3^{NB} do not cause their respective normality concepts to be inconsistent, either; consequently, both translations are included in Π .

In Phase 2, the algorithm realizes at line 17 that δ_2^{NA} has been erroneously discarded and inserts $NA \sqsubseteq \perp$ in \mathcal{KB}^* . However, \mathcal{KB}^* contains also Π (line 11), hence δ_3^{NB} . It follows that $\mathcal{KB}^* \models NB \sqsubseteq \perp$.

Summarizing, when \mathcal{KB} is *not* N-free, Algorithm 1 may erroneously insert in $\Pi \subseteq \mathcal{KB}^*$ the translation of some DIs that should be overridden due to DIs that are temporarily discarded in Phase 1. \square

¹⁰The reader may easily verify that the final result is the same for the other possible linearization, i.e. $\delta_2, \delta_1, \delta_3$.

Despite this limitation, we expect the optimistic evaluation method to be useful in a non-negligible amount of cases. So far, the only known application of normality concepts in knowledge base axioms is for asserting that some attributes of normal individuals are normal, too, as in the next example.

Example 9 ([15]). The heart of most humans is on the left-hand side of the body, but there are exceptions (so-called *situs inversus*). Thus, the standard position of the human heart should be axiomatized as a defeasible property. For this purpose, one can first describe normal human hearts, and then assert that typical humans have normal human organs:

$$\begin{aligned}
\text{HumanHeart} &\equiv \text{HumanOrgan} \sqcap \text{Heart} , \\
\text{Human} &\sqsubseteq \exists \text{has_organ}.\text{Heart} , \\
\text{HumanHeart} &\sqsubseteq_n \exists \text{has_position}.\text{Left} , \\
\text{Human} &\sqsubseteq_n \forall \text{has_organ}.\text{NHumanOrgan} .
\end{aligned} \tag{46}$$

Note the explicit use of NHumanOrgan in (46) to state that the organs of normal humans are normal themselves. These axioms entail that the heart of typical humans is placed on the left-hand side of the body, that is:

$$\mathcal{KB} \approx \text{NHuman} \sqsubseteq \exists \text{has_organ} . (\text{Heart} \sqcap \exists \text{has_position} . \text{Left}) .$$

□

It is not always required to state that role values are normal, though; see the examples in Section 2.3, and also Example 11 in [15], that illustrates a case where stating that role values are normal is not even correct. Moreover, N-module extraction may eliminate all occurrences of normality concepts, and return an N-free module. Thus, N-free knowledge bases are likely to occur in practice, for different reasons. In the section devoted to performance analysis we will see that the optimistic evaluation method (when applicable) brings tangible benefits, especially when combined with module extraction.

4. System Implementation

We introduce `NMReasoner`, a \mathcal{DL}^N implementation based on existing classical reasoners. Since mainstream DL technology does not support nonmonotonic inferences and the available implementations of nonmonotonic DLs can only handle knowledge bases with moderate size, currently, no “real” nonmonotonic DL

knowledge bases exist. Thus, in order to analyze the reasoner’s performance experimentally, in Section 4.2 we propose a systematic approach to the automatic generation of synthetic test cases with realistic size and structure.

4.1. Reasoner Implementation

NMReasoner is implemented in Java 8 and distributed as a .jar file.

According to the theoretical framework, the engine consists of two modules. The first one, hereafter called *translation module*, constructs the classical knowledge base \mathcal{KB}^Σ corresponding to the given \mathcal{DL}^N knowledge base $\mathcal{KB} = \mathcal{S} \cup \mathcal{D}$ (where \mathcal{S} and \mathcal{D} are the sets of strong and defeasible axioms, respectively). The second module computes nonmonotonic subsumptions, sometimes called *queries* in the following. Besides \mathcal{KB} , NMReasoner takes as input the set Σ of all normality concepts that occur either in \mathcal{KB} or in a given set of queries Q . Moreover, if no priority relation over DIs is provided as part of the input (encoded in an appropriate file), then specificity (8) is applied by default. Both modules call an external classical reasoner for classification. In particular, we use the specialized engine ELK [42, 43, 44] with knowledge bases that belong to \mathcal{EL} , and the general engine HerMiT [45] otherwise.

NMReasoner implements the optimizations described in Section 3. By default, all optimisations are turned off and the reasoner relies only on the optimization techniques supported by the underlying, classical reasoning engine. The usage of appropriate input parameters enables selectively one or more ad hoc optimisations.

4.2. Test Suites Generator

The test suite generator takes as input a classical knowledge base \mathcal{S} , which serves as basic pattern, and returns a \mathcal{DL}^N knowledge base according to the following steps.

Step 1: DI generation. DIs are generated by either injecting fully synthetic DIs in \mathcal{S} , or transforming a random set of CIs occurring in \mathcal{S} into DIs. A fully synthetic DI takes the form $A \sqsubseteq_n B$ or $A \sqsubseteq_n \exists R.B$ where the atomic concepts A and B , and the role R are randomly chosen from the signature of \mathcal{S} . The generator makes sure that no duplicates are generated, and that for each new DI δ , $\text{pre}(\delta) \sqsubseteq \text{con}(\delta)$ is not classically entailed by \mathcal{S} . The number of fully synthetic and transformed DIs are respectively tuned by the Synthetic-DI-rate and CI-to-DI-rate parameters (see Figure 1).

Step 2: Disjointness axiom injection. In order to increase the probability of overriding (and hence nonmonotonic behavior), some additional inconsistencies between DI conclusions are injected. To do that, DI pairs (δ_1, δ_2) are randomly

Parameter	Meaning
Synthetic-DI-rate	% of fully synthetic DIs w.r.t. the number of CIs
CI-to-DI-rate	% of CIs to be transformed DIs
DA-rate	% of disjointness axioms w.r.t. the number of CIs
I-rate	% of new distinct individuals w.r.t. the number of concept names
ABox-rate	% of assertions w.r.t. the number of concept names
R-rate	% of role assertions w.r.t. the number of all assertions
NC-size	# of DIs with normality concept within the scope of quantifiers

Figure 1: The main parameters of the test-case generator. All parameters, except NC-size, are represented as a rate of the number of CIs occurring in \mathcal{S} .

selected and, for each of them, two arbitrary concepts C_1 and C_2 are picked from the superclasses of $\text{con}(\delta_1)$ and $\text{con}(\delta_2)$. Then, new disjointness axioms $C_1 \sqcap C_2 \sqsubseteq \perp$ are added to \mathcal{S} . The test case generator makes sure that none of C_1 , C_2 , and \mathcal{S} are made inconsistent, by checking that the following conditions are satisfied in the extended knowledge base: (i) $\text{con}(\delta_2) \not\sqsubseteq C_1$; (ii) $\text{con}(\delta_1) \not\sqsubseteq C_2$; (iii) $C_2 \not\sqsubseteq C_1$ and (iv) $C_1 \not\sqsubseteq C_2$. Note that we intentionally refrain from asserting $\text{con}(\delta_1) \sqcap \text{con}(\delta_2) \sqsubseteq \perp$ directly, so that the reasoning involved in checking whether a DI is overridden is generally nontrivial. The generation of disjointness axioms is controlled by the DA-rate parameter.

Step 3: N-transformation. Normality concepts are introduced by randomly choosing a DI $A \sqsubseteq_n \exists R.B$ and replacing it with $A \sqsubseteq_n \exists R.N B$.¹¹ The amount of normality concept occurrences is controlled by parameter *NC-size* specifying the number of modified DIs.

The appropriateness of the test suites generated with the above procedure is assessed by inspecting the structure of the generated knowledge bases. For the sake of readability, this ancillary assessment is reported in the appendix.

Queries have the form $NA \sqsubseteq B$ where A and B are randomly selected among the concepts defined in the knowledge base. Such queries check whether the normal instances of A have property B .

¹¹ So far, all the application examples that are not N-free satisfy this restriction, as apparently the only purpose of explicit normality concepts is restricting default role ranges to normal individuals.

5. Experimental Evaluation

In this section we present a performance analysis of NMReasoner. All test cases result from our test suites generator, taking as input one of the following ontologies: a version of the Gene Ontology¹² (GO for short) published in 2006, Fly Anatomy (FLY)¹³ and SNOMED CT (Systematized Nomenclature of Medicine Clinical Terms)¹⁴. These ontologies have been extensively used in many performance experiments [22, 23, 24, 25, 26, 27, 28]. They are particularly well-suited to our purposes because they fit well our application scenarios, being large biomedical ontologies. In particular, GO features 20465 atomic concepts and 28896 concept inclusions; FLY has 7797 atomic concepts and 19137 concept inclusions; and SNOMED CT has 291151 atomic concepts, 227704 concept inclusions and 63452 concept equivalences. They can be encoded in \mathcal{EL}^{++} , whose nonmonotonic version \mathcal{EL}^{++N} enjoys tractable inference problems.

Hereafter, we refer to the optimization techniques introduced in Section 3 with the following labels:

Naive: A naive reasoner which realizes the classical reduction in Section 2.3 *as such*, without any optimization technique besides those natively supported by the underlying classical reasoner. We use this reasoner as a baseline in our performance analysis.

Opt: The optimistic method introduced in Sec. 3.2.

Mod: The N-module extraction method of Sec. 3.1, that uses the module extraction facility of the OWL API.

MO: The sequential execution of Mod and Opt. This combined method is correct for N-free knowledge bases by Theorem 12 and Theorem 10.

Mod*: The iterated N-module extraction of Sec. 3.1

In order to analyze the contribution of each optimization technique on performance, we designed specific test suites corresponding to different settings of the

¹²<http://www.geneontology.org>

¹³One of the largest ontologies listed at the OBO Foundry websites <http://www.obofoundry.org/>.

¹⁴The largest biomedical ontology obtained from the official July 2015 international release by converting from the native syntax (RF2) to FSS using the supplied converter.

parameters in Figure 1. First, we consider a test suite where only N-free defeasible inclusions are added to the input ontologies (recall that the optimistic method is applicable only to N-free knowledge bases). Note that, since this test suite is N-free, Mod is idempotent and Mod* is equivalent to Mod. Consequently, we use this test suite to evaluate only Opt, Mod, and their combination MO. Mod* will be assessed with a second test suite, by gradually injecting normality concepts.

Every single value that is reported in the following sections is obtained as the average execution time over five randomly generated nonmonotonic ontologies and fifty different queries on each ontology. Execution time includes the translation into classical DL of \mathcal{KB} or – if module extraction is applied – of an N-module \mathcal{M} of \mathcal{KB} . The experiments were performed on a server with an 8-cores processor Intel Xeon Silver 4110, 11M cache, 198GB RAM, running Ubuntu 18.04. NMReasoner was run on Java 1.8, configured with 12GB heap memory and 4GB stack memory space. We have not exploited parallelism.

label	CI-to-DI-rate	Synthetic-DI-rate	DA-rate
PS1	5%, 10%, 15%, 20%, 25%	0%	15%
PS2	0%	5%, 10%, 15%, 20%, 25%	15%
PS3	15%	0%	5%, 10%, 15%, 20%, 25%
PS4	0%	15%	5%, 10%, 15%, 20%, 25%

Figure 2: Parameter settings used to evaluate Opt, Mod, and MO.

5.1. Module extraction vs. optimistic computation

In this section we compare the effectiveness of optimization techniques Opt, Mod, and their combination MO. To this end, we consider a test suite where the input ontologies are augmented by N-free DIs only (as the optimistic method is correct only for such knowledge bases). More specifically, the suite is composed by 4 parameter settings where only one of CI-to-DI-rate, Synthetic-DI-rate, and DA-rate is allowed to vary (see Figure 2). The parameters not mentioned in Figure 2 are set to 0, consequently Σ contains only the normality concept occurring in the current query. Settings PS1 and PS2 show the impact of $|\mathcal{D}|$ (the number of DIs) on performance, while PS3 and PS4 evaluate the cost of processing disjointness axioms. In PS1 and PS3, DIs are generated by converting strong CIs, whereas PS2 and PS4 inject fully synthetic DIs.

Note that PS1 and PS3 preserve the overall number of (strong and defeasible) inclusions, while PS2 and PS4 increase the size of the knowledge base up to 25%;

therefore the experiments based on fully synthetic DIs are expected to take longer computation time.

Figure 3 reports the execution times for the test cases PS1 and PS2. In both cases, the methods Naive and Opt grow linearly with $|\mathcal{D}|$. However, Opt is about two times faster in GO, FLY and SNOMED CT. Note that response times are slower in PS2, since the generated knowledge bases are up to 25% larger in this setting, as explained above.

Mod turns out to be far more effective than Opt. On average, in PS1, it is approximately 85 times faster than Naive for GO and FLY (max. speedup is 146), and 3500 times faster for SNOMED CT. In PS2, Mod is 15 times faster for GO and FLY (max. speedup 33), and 70 times faster for SNOMED CT.

On average, the combination of module extraction and optimistic computation (MO) yields in PS2 a further 45% improvement over Mod alone; the maximum reduction (63%) is obtained in SNOMED with Synthetic-DI-rate=25%, and DA-rate=15% (Figure 3, last line in the rightmost table). On the other hand, in PS1, the improvement of MO over Mod alone is barely perceptible. This is due to the fact that the optimistic method is particularly effective on large modules, while on small modules ($< 10^2$ axioms) it brings little benefits, and its overhead may occasionally slow down the computation. In particular, the modules produced with PS2 are larger because synthetic DIs connect unrelated parts of the ontology. Most modules have $10^5 - 10^6$ axioms; in this range, the benefits of the optimistic computation prevail and MO notably outperforms Mod. In PS1, instead, Mod returns middle-sized modules with approximately $10^2 - 10^3$ axioms. This is a gray area where the benefits of the optimistic algorithm are more sensitive to the structure of the ontology and the query.

The additional conflicts induced by injected disjointness axioms have moderate effects on execution time (cf. Figure 4, dedicated to settings PS3 and PS4). This should be expected since the number of classical entailments needed to translate the knowledge base in classical \mathcal{DL} depends on the number of inclusions δ_i^{NC} that must be processed, which is not affected by disjointness axioms. Such axioms increase the size of the knowledge base, but are processed efficiently by modern classical reasoners (and ELK in particular). Mod’s average response time across both PS3 and PS4 is 0.89 sec. for Gene Ontology, 1.41 sec. for Fly Anatomy and 111.28 for SNOMED CT, while the longest Mod response times have been 1.6 sec., 2.82 sec. and 116 sec. respectively. In all cases, the speedups of Mod and MO over Naive remain well above one order of magnitude.

CI-to-DI	Naive	Opt	Mod	MO	Synth-DI	Naive	Opt	Mod	MO
GO					GO				
05%	12.3	4.99	0.26	0.24	05%	13.1	05.17	0.48	0.42
10%	24.1	9.49	0.28	0.26	10%	27.8	09.85	0.83	0.64
15%	34.5	14.3	0.29	0.29	15%	37.5	15.5	1.47	0.98
20%	42.0	19.7	0.32	0.31	20%	46.1	21.2	2.76	1.54
25%	49.9	24.9	0.34	0.33	25%	57.1	27.6	4.66	2.46
FLY					FLY				
05%	4.22	1.90	0.13	0.12	05%	4.86	02.01	0.40	0.35
10%	7.97	3.78	0.15	0.14	10%	9.86	03.97	1.19	0.67
15%	11.9	5.60	0.17	0.16	15%	14.7	06.01	2.51	1.18
20%	14.4	7.34	0.19	0.18	20%	19.9	08.59	4.61	2.15
25%	17.5	9.18	0.21	0.20	25%	24.3	10.8	7.25	3.34
SNOMED					SNOMED				
05%	2630	1191	1.52	1.49	05%	2843	1113	27.1	11.0
10%	4389	2384	1.73	1.68	10%	4808	2117	51.3	22.5
15%	6844	3782	1.91	1.85	15%	8139	3828	112	51.7
20%	9925	5260	2.24	2.18	20%	11776	5732	248	96.8
25%	13638	6158	2.61	2.43	25%	15094	6637	521	195

Figure 3: Impact of $|\mathcal{D}|$ on performance (sec). Parameter settings PS1 (left) and PS2 (right).

DA	Naive	Opt	Mod	MO	DA	Naive	Opt	Mod	MO
GO					GO				
05%	27.4	13.3	0.28	0.27	05%	33.3	13.7	1.26	0.86
10%	27.5	13.8	0.29	0.28	10%	35.6	14.7	1.35	0.92
15%	34.5	14.3	0.29	0.29	15%	37.5	15.5	1.47	0.98
20%	38.6	15.0	0.30	0.29	20%	42.0	16.7	1.51	1.00
25%	36.2	15.6	0.31	0.30	25%	43.0	16.9	1.60	1.06
FLY					FLY				
05%	10.3	5.02	0.16	0.15	05%	12.9	5.55	2.23	1.07
10%	11.5	5.38	0.17	0.16	10%	14.0	5.84	2.50	1.14
15%	11.9	5.60	0.17	0.17	15%	14.8	6.01	2.52	1.18
20%	12.1	5.85	0.18	0.17	20%	15.9	6.30	2.61	1.22
25%	12.6	6.20	0.19	0.18	25%	16.6	6.59	2.82	1.30
SNOMED					SNOMED				
05%	6617	3689	1.87	1.82	05%	7721	3517	104	47.5
10%	6872	3795	1.90	1.85	10%	7895	3749	107	49.2
15%	6844	3782	1.91	1.85	15%	8139	3828	112	51.7
20%	7344	3881	2.01	1.90	20%	8243	3899	114	56.0
25%	7924	3901	2.02	1.91	25%	8665	4065	116	62.9

Figure 4: Impact of DAs on performance (sec). Parameter settings PS3 (left) and PS4 (right).

5.2. Effectiveness of Iterated Module Extraction

In the following we analyse the performance of the iterated module extractor Mod^* by comparing it with the non-iterated version Mod . Here we use knowledge bases that contain occurrences of normality concepts, because – as explained above – on N-free inputs Mod^* and Mod return the same results. Since the input knowledge bases are not N-free, Opt is not applicable to this test set.

Mod^* shows its benefits as soon as the number of normality concepts in the knowledge base (and in the relevant signature Σ) increases. Figure 5 illustrates a test suite where normality concepts are injected in \mathcal{KB} by tuning the parameter NC-size (see Step 3 in Section 4.2). This parameter sets the size of Σ from 50 to 250 normality concepts, while DA-rate is set to the intermediate value 15%, and both CI-to-DI-rate (Figure 5, left) and Synthetic-DI-rate (Figure 5, right) are set to 25%.

With the Naive method, the translation of each DI δ must process an inclusion δ^{NC} , for all $\text{NC} \in \Sigma$. Thus, the computational cost of Naive rapidly increases with $|\Sigma|$. The execution time always exceeds the timeout (30 min), and is not reported in Figure 5.

With Mod all the injected normality concepts are included in the relevant signature Γ (cf. Table 2), so the extracted modules are quite large and the computation time is considerably slower, compared to the N-free tests in Section 5.1. The iterated module extraction Mod^* makes \mathcal{DL}^{N} reasoning more than one order of magnitude faster than Mod . This remarkable result can be explained by the twofold benefits of iteratively reducing the size of the relevant signature Γ : First, if at some iteration a normality concept NC is removed from the current module, then the next iteration of Mod^* starts with a smaller Γ and has the opportunity of further reducing the module’s size. Moreover, translating a smaller final module means – in the best case – processing a quadratically smaller number of inclusions δ^{NC} , since both the set of DIs δ and the set of NC occurring in the final module decrease.

5.3. Some Remarks on Modules not Based on Locality

We conclude this section with a remark on module extractors based on principles different from locality, such as PrisM for example [46]. In our algorithms, locality-based module extractors cannot be simply replaced with arbitrary module extractors, because the proof of Theorem 10 (on which the correctness of our approach is based) relies on locality in many points. For example, it is not hard to

$ \Sigma $	Mod	Mod*
GO		
50	2.70	0.45
100	8.59	0.45
150	16.9	0.46
200	28.2	0.46
250	42.0	0.46
FLY		
50	10.4	0.47
100	23.4	0.58
150	42.8	0.66
200	64.6	0.73
250	88.0	0.99
SNOMED		
50	8.79	4.82
100	26.7	4.77
150	44.9	5.32
200	83.8	5.37
250	118	4.64

$ \Sigma $	Mod	Mod*
GO		
50	186	8.18
100	414	10.4
150	697	15.3
200	1012	20.4
250	1412	28.3
FLY		
50	288	22.4
100	620	34.0
150	1020	47.8
200	1478	63.5
250	2103	83.5
SNOMED		
50	16098	1004
100	34628	2604
150	40996	3565
200	63696	5398
250	83782	6362

Figure 5: Non N-free tests. Impact of normality concepts (sec). DA rate = 15%. CI-to-DI-rate = 25% (left), Synthetic-DI = 25% (right).

see that Lemma 4 cannot be extended to arbitrary modules \mathcal{M} , even if \mathcal{KB} is classic. Consequently, it is not yet clear if and how different notions of modules can be used to optimize reasoning in \mathcal{DL}^N . The answer to this question lies beyond the scope of this paper and is an interesting subject for future work.

6. Related Work on Optimizations for Nonmonotonic DL

Module extraction is used for speeding up reasoning in several works on non-monotonic DLs based on *rational closure* [20, 47]. To see the details, let us sketch the computation of the rational closure of a knowledge base \mathcal{KB} . In a preliminary phase, a finite sequence $\langle \mathcal{E}_i \rangle_i$ of subsets of \mathcal{KB} is computed, such that:

$$\mathcal{KB} \supseteq \mathcal{E}_1 \supseteq \mathcal{E}_2 \supseteq \dots \supseteq \mathcal{E}_n.$$

A concept C has *rank* i iff

$$\mathcal{E}_i \models \top \sqsubseteq_n \neg C \text{ and } \mathcal{E}_{i+1} \not\models \top \sqsubseteq_n \neg C; \quad (47)$$

if no such i exists, then the rank of C is ∞ . Then, for each query $C \sqsubseteq_n D$, the answer is “yes” iff the rank of C is either ∞ or smaller than the rank of $C \sqcap \neg D$. The computation of the rank of C and $C \sqcap \neg D$ can be accelerated by extracting a module \mathcal{M} of \mathcal{KB} w.r.t. the signature of $C \sqcap \neg D$, and modifying (47) by intersecting \mathcal{E}_i and \mathcal{E}_{i+1} with \mathcal{M} . The experiments reported in [20, 47] have been run on small, randomly generated knowledge bases in \mathcal{ALC} , with a number of axioms that ranges uniformly from 150 to 5150. On these ontologies, the computation of the ranking takes between a few milliseconds and approximately 24 seconds, depending on the ontology size and number of ranks, while the average query answering time is less than 2 ms.

Another nonmonotonic extension of DLs based on rational closure, featuring an explicit typicality operator, has been implemented and compared with the naive implementation of \mathcal{DL}^N [48]. The comparison is carried out using the N-free test sets obtained from GO with the CI-to-DI transformation method. No optimization is applied. The computation of the ranking ranges approximately between 3.5 minutes and 2.5 hours, while query time ranges between 8.87 sec and 81.15 sec. More precisely, query response times lie between those of our naive implementation and those of Opt (optimistic computation alone), so Mod and the combined method are significantly faster.

Nonmonotonic description logics are related to Datalog[±] with nonmonotonic negation. Datalog[±] admits existential quantification in rule heads, and in this way

can encode concept inclusions that have existential restrictions in the right-hand side. In general, the algorithms for reasoning in Datalog^\pm are not guaranteed to terminate, so several restrictions on rule bases have been introduced to enforce termination and upper complexity bounds. Semantics, decidability and complexity of Datalog^\pm with nonmonotonic negation have been extensively studied, see for example [49, 50, 51, 52, 53, 54, 55]. As far as we know, implementations are not currently available. The implementations of Datalog^\pm and its extensions sometimes adopt optimization techniques developed for deductive databases, such as magic sets, that – like module extraction – have the purpose of focussing reasoning on the symbols relevant to the query. Magic sets have been applied both to the implementation of (monotonic) Datalog^\pm [56], and to the implementation of Datalog with nonmonotonic negation and aggregates [57]. Therefore, magic sets constitute an appealing candidate for optimized implementations of nonmonotonic Datalog^\pm .

7. Conclusions

In this paper we have introduced optimization techniques for nonmonotonic reasoning in \mathcal{DL}^N . Our experimental results (summarized below) show that, with such optimizations, nonmonotonic reasoning becomes feasible for large and very large real-world knowledge bases. To the best of our knowledge this is the first work – in the area of description logics – where nonmonotonic reasoning is applied to knowledge bases with a number of axioms ranging between $\sim 20,000$ and $\sim 300,000$.

First consider the nonmonotonic versions of FLY ($>19\text{K}$ axioms) and GO ($>28\text{K}$ axioms) obtained by transforming classical inclusions into defeasible inclusions (CI-to-DI method). In all of these experiments the average query response time is below one second, and mostly below 0.5 seconds. Similar experiments based on SNOMED ($>290\text{K}$ axioms) yield response times between 1.5 and 4.6 seconds, that are compatible with a wide range of use cases.

In another set of experiments, we have injected random DIs in FLY, GO, and SNOMED, thereby increasing their size up to 25%, and introducing random dependencies between different parts of the knowledge base. For the smaller ontologies FLY and GO, response times are always below 1.5 minutes, and less than 4 seconds in most test cases. For SNOMED, the hardest test cases ($\sim 72\text{K}$ additional axioms, 250 normality concepts in the knowledge base) took approximately 1 hour and 45 minutes, while the simplest test cases (N-free) are completed within approximately 1 minute.

Summarizing, the results for \mathcal{DL}^N ontologies up to approximately 35K axioms show that the available technology can process subsumption queries with response times that are compatible with many practical applications, including interactive query answering. For the knowledge bases derived from SNOMED, that are one order of magnitude larger, response times may vary depending on the factors that most significantly affect performance, such as the number of normality concepts occurring in the knowledge bases and the amount of logical dependencies between different concepts. In the test cases where the size and the structure of SNOMED are preserved, response times are compatible with interactive query answering, while response times in fully random test cases range from 11 seconds to almost two hours.

The above results have been obtained with the following optimizations. The most general and effective optimizations consists of a module extractor for \mathcal{DL}^N (Mod for short), that can be effectively used to focus reasoning on a subset of the knowledge base that is relevant to the given query. Compared with the naive reasoner published in [15], the optimized version is at least one order of magnitude faster, and up to $\sim 10^3$ times faster in some test cases.

Typically, when no role fillers are restricted to normal individuals, the knowledge base is N-free (i.e. it does not contain any occurrence of normality concepts). For N-free knowledge bases, we developed a second optimization technique, called optimistic computation (Opt), that is a heuristics aimed at reducing the number of retractions in the reasoning algorithm. The combination of optimistic computation with module extraction (MO), on average, brings a 23% speedup over module extraction alone.

When the knowledge base is not N-free, the optimistic method is not applicable. However, in this case, the extractor for \mathcal{DL}^N is not idempotent and can be profitably iterated (Mod*). The resulting response times are well over one order of magnitude faster than the non-iterated version, and this makes it possible to process large, non-N-free knowledge bases in practice.

The problem of defining correct module extraction techniques for nonmonotonic logics is still open, in general. Even within the scope of \mathcal{DL}^N , our proofs do not cover module extraction techniques based on principles different from locality. Moreover, the possible applications of module extraction span beyond optimization. A general study of module extraction for nonmonotonic reasoning is an interesting subject for further research.

Appendix

Test case Structure

The randomly generated test cases have been analyzed as follows, to make sure that they are not trivial. For each normality concept in Σ we measure the height of the priority hierarchy of applicable DIs, and how many applicable DIs are overridden; the former quantity is related to the potential levels of overriding, while the latter is more directly related to the nonmonotonic behavior that actually occurs.

A structural analysis of the test cases coming from Gene Ontology, Fly and SNOMED CT is shown in Figures 6, 7, and 8, respectively. As the DA rate grows, the figures report the percentage of overridden applicable DIs, plus the average and maximum height of the priority hierarchy of applicable DIs (hDIs, for brevity). The tables on the left are devoted to DIs obtained from strong inclusions (CI-to-DI-rate=15%) whereas the ones on the right refers to fully synthetic DIs (Synthetic-DI-rate=15%). For all figures, every single reported value is obtained as the average over ten different non monotonic ontologies and fifty different queries on each ontology, each of which involved the construction of a different translation \mathcal{KB}^Σ (as Σ depends on the query).

As expected, each increment of the DA-rate (hence of disjointness axioms) causes an increment of the percentage of overridden DIs. This relation is relatively less evident for synthetic DIs, probably due to the further randomness introduced by fully random DI generation. The amount of overriding is reasonable: the percentage of overridden DIs ranges from 61.9% to 87.6% for GO, from 34.4% to 66.0% for FLY, and from 29.3% to 66.9% for SNOMED, depending on the amount of disjointness axioms (hence DI conflicts) contained in the KB.

Furthermore, the reported results fit well the structure of GO, FLY and SNOMED. The length of the longest path in GO's classification (i.e. the maximum possible hierarchy depth) is 15, and the average length 3.66. In the experiments concerning DIs obtained from CIs (see Figure 6, left) the average (resp. maximal) depth of the applicable DIs hierarchy range between 41% and 55% of the average (resp. 26,6% and 40% of the maximal) path length in the original ontology, coherently with the random placement of normality concepts in the hierarchy. For fully synthetic DIs (Figure 6, right) these values are lower and vary between 34% and 42% of the average length, and between 20% and 33,3% of the maximum length.

Analogously, the length of the longest path in FLY's classification is 16 (resp. 27 in SNOMED's classification), and the average length 6.78 (resp. 9.02). The depth of the DI priority hierarchy (which is related to the interference between different DIs and, indirectly, the number of exception levels) ranges between 19%

DA-rate	Overr./Appl. DIs	Avg. (Max) hDIs	DA-rate	Overr./Appl. DIs	Avg. (Max) hDIs
5%	61,86%	2,03 (6)	5%	83,55%	1,34 (4)
10%	70,71%	1,96 (6)	10%	84,13%	1,40 (4)
15%	68,45%	2,01 (5)	15%	91,17%	1,25 (3)
20%	75,63%	1,71 (5)	20%	91,47%	1,30 (4)
25%	80,28%	1,54 (4)	25%	87,64%	1,50 (4)

Figure 6: Values characterizing the experiments with variable DA-rate in GO. CI-to-Di-rate=15% (left), Synthetic-DI-rate=15% (right).

DA-rate	Overr./Appl. DIs	Avg. (Max) hDIs	DA-rate	Overr./Appl. DIs	Avg. (Max) hDIs
5%	34,44%	2,26 (8)	5%	37,95%	2,17 (5)
10%	42,6%	2,6 (7)	10%	55,56%	1,74 (3)
15%	39,1%	2,55 (6)	15%	57,03%	1,81 (4)
20%	46,4%	2,44 (7)	20%	57,32%	1,95 (5)
25%	37,53%	2,49 (7)	25%	65,98%	1,83 (3)

Figure 7: Values characterizing the experiments with variable DA-rate in FLY. CI-to-Di-rate=15% (left), Synthetic-DI-rate=15% (right).

and 50% of the depth of FLY's taxonomy and 15% and 44% of the depth of SNOMED's taxonomy.

References

- [1] A. L. Rector, Defaults, context, and knowledge: Alternatives for OWL-indexed knowledge bases, in: Pacific Symposium on Biocomputing, World Scientific, 2004, pp. 226–237.
- [2] R. Stevens, M. E. Aranguren, K. Wolstencroft, U. Sattler, N. Drummond, M. Horridge, A. L. Rector, Using OWL to model biological knowledge, International Journal of Man-Machine Studies 65 (7) (2007) 583–594.
- [3] P. A. Bonatti, P. Samarati, Logics for authorization and security, in: Logics for Emerging Applications of Databases, Springer, 2003, pp. 277–323.
- [4] P. A. Bonatti, M. Faella, L. Sauro, Adding default attributes to EL++, in: W. Burgard, D. Roth (Eds.), AAI, AAI Press, 2011.

DA-rate	Overr./Appl. DIs	Avg. (Max) hDIs	DA-rate	Overr./Appl. DIs	Avg. (Max) hDIs
5%	29,27%	6,26 (12)	5%	48,38%	3,17 (6)
10%	33,49%	5,61 (9)	10%	53,92%	2,74 (4)
15%	39,19%	6,55 (10)	15%	57,03%	2,81 (5)
20%	45,44%	6,34 (10)	20%	58,28%	2,95 (5)
25%	46,08%	6,79 (11)	25%	66,91%	2,38 (6)

Figure 8: Values characterizing the experiments with variable DA-rate in SNOMED. CI-to-Di-rate=15% (left), Synthetic-DI-rate=15% (right).

- [5] P. A. Bonatti, M. Faella, L. Sauro, EL with default attributes and overriding, in: *Int. Semantic Web Conf. (ISWC 2010)*, Vol. 6496 of LNCS, Springer, 2010, pp. 64–79.
- [6] P. A. Bonatti, M. Faella, L. Sauro, Defeasible inclusions in low-complexity DLs, *J. Artif. Intell. Res. (JAIR)* 42 (2011) 719–764.
- [7] F. M. Donini, D. Nardi, R. Rosati, Autoepistemic description logics, in: *IJCAI (1)*, 1997, pp. 136–141.
URL citeseer.ist.psu.edu/donini97autoepistemic.html
- [8] F. M. Donini, D. Nardi, R. Rosati, Description logics of minimal knowledge and negation as failure, *ACM Trans. Comput. Log.* 3 (2) (2002) 177–225.
- [9] L. Giordano, V. Gliozzi, N. Olivetti, G. Pozzato, Reasoning about typicality in preferential description logics, in: *Proc. of Logics in Artificial Intelligence, 11th European Conference, JELIA 2008*, Vol. 5293 of Lecture Notes in Computer Science, Springer, 2008.
- [10] L. Giordano, V. Gliozzi, N. Olivetti, G. L. Pozzato, A non-monotonic description logic for reasoning about typicality, *Artif. Intell.* 195 (2013) 165–202. doi:10.1016/j.artint.2012.10.004.
URL <https://doi.org/10.1016/j.artint.2012.10.004>
- [11] L. Giordano, N. Olivetti, V. Gliozzi, G. L. Pozzato, ALC + T: a preferential extension of description logics, *Fundam. Inform.* 96 (3) (2009) 341–372.
- [12] L. Giordano, V. Gliozzi, N. Olivetti, G. L. Pozzato, Semantic characterization of rational closure: From propositional logic to description logics, *Artif.*

- Intell. 226 (2015) 1–33. doi:10.1016/j.artint.2015.05.001.
URL <https://doi.org/10.1016/j.artint.2015.05.001>
- [13] G. Casini, U. Straccia, Rational closure for defeasible description logics, in: T. Janhunen, I. Niemelä (Eds.), JELIA, Vol. 6341 of Lecture Notes in Computer Science, Springer, 2010, pp. 77–90.
- [14] P. A. Bonatti, Rational closure for all description logics, *Artif. Intell.* 274 (2019) 197–223. doi:10.1016/j.artint.2019.04.001.
URL <https://doi.org/10.1016/j.artint.2019.04.001>
- [15] P. A. Bonatti, M. Faella, I. Petrova, L. Sauro, A new semantics for overriding in description logics, *Artif. Intell.* 222 (2015) 1–48. doi:10.1016/j.artint.2014.12.010.
URL <http://dx.doi.org/10.1016/j.artint.2014.12.010>
- [16] P. A. Bonatti, L. Sauro, On the logical properties of the nonmonotonic description logic DL^N , *Artif. Intell.* 248 (2017) 85–111.
- [17] P. A. Bonatti, I. M. Petrova, L. Sauro, Optimizing the computation of overriding, in: *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference*, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I, 2015, pp. 356–372.
- [18] M. Pensel, A lightweight defeasible description logic in depth, Ph.D. thesis, TU Dresden (2019).
- [19] G. Casini, U. Straccia, T. Meyer, A polynomial time subsumption algorithm for nominal safe ELO^\perp under rational closure, *Inf. Sci.* 501 (2019) 588–620. doi:10.1016/j.ins.2018.09.037.
URL <https://doi.org/10.1016/j.ins.2018.09.037>
- [20] G. Casini, T. Meyer, K. Moodley, I. J. Varzinczak, Towards practical defeasible reasoning for description logics, in: T. Eiter, B. Glimm, Y. Kazakov, M. Krötzsch (Eds.), *Description Logics*, Vol. 1014 of CEUR Workshop Proceedings, CEUR-WS.org, 2013, pp. 587–599.
- [21] G. Casini, U. Straccia, Defeasible inheritance-based description logics, *J. Artif. Intell. Res. (JAIR)* 48 (2013) 415–473.

- [22] F. Baader, C. Lutz, B. Suntisrivaraporn, CEL - a polynomial-time reasoner for life science ontologies, in: U. Furbach, N. Shankar (Eds.), IJCAR, Vol. 4130 of Lecture Notes in Computer Science, Springer, 2006, pp. 287–291.
- [23] V. Delaitre, Y. Kazakov, Classifying ELH ontologies in SQL databases, in: OWLED, 2009.
- [24] J. Mendez, B. Suntisrivaraporn, Reintroducing cel as an owl 2 el reasoner, in: B. C. Grau, I. Horrocks, B. Motik, U. Sattler (Eds.), Proceedings of the 2009 International Workshop on Description Logics (DL2009), Vol. 477 of CEUR-WS, 2009.
- [25] B. Glimm, I. Horrocks, B. Motik, R. Shearer, G. Stoilos, A novel approach to ontology classification, *J. Web Sem.* 14 (2012) 84–101.
- [26] Y. Kazakov, Consequence-driven reasoning for Horn SHIQ Ontologies, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI’09, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009, pp. 2040–2045.
URL <http://dl.acm.org/citation.cfm?id=1661445.1661771>
- [27] B. Sertkaya, In the search of improvements to the EL+ classification algorithm, in: Description Logics, 2011.
- [28] D. Tsarkov, I. Horrocks, P. F. Patel-Schneider, Optimizing terminological reasoning for expressive description logics, *J. Autom. Reasoning* 39 (3) (2007) 277–316.
- [29] I. Horrocks, O. Kutz, U. Sattler, The even more irresistible SROIQ, in: P. Doherty, J. Mylopoulos, C. A. Welty (Eds.), Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006, AAAI Press, 2006, pp. 57–67.
URL <http://www.aaai.org/Library/KR/2006/kr06-009.php>
- [30] F. Baader, D. L. McGuinness, D. Nardi, P. Patel-Schneider, The Description Logic Handbook: Theory, implementation and applications, Cambridge University Press, 2003.

- [31] S. Rudolph, Foundations of description logics, in: A. Polleres, C. d'Amato, M. Arenas, S. Handschuh, P. Kroner, S. Ossowski, P. F. Patel-Schneider (Eds.), Reasoning Web. Semantic Technologies for the Web of Data - 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures, Vol. 6848 of Lecture Notes in Computer Science, Springer, 2011, pp. 76–136. doi:10.1007/978-3-642-23032-5_2.
URL https://doi.org/10.1007/978-3-642-23032-5_2
- [32] U. Sattler, T. Schneider, M. Zakharyashev, Which kind of module should I extract?, in: Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009, 2009.
URL http://ceur-ws.org/Vol-477/paper_33.pdf
- [33] C. Lutz, F. Wolter, Deciding inseparability and conservative extensions in the description logic EL, *J. Symb. Comput.* 45 (2) (2010) 194–228. doi:10.1016/j.jsc.2008.10.007.
URL <https://doi.org/10.1016/j.jsc.2008.10.007>
- [34] R. Kontchakov, F. Wolter, M. Zakharyashev, Can you tell the difference between DL-Lite ontologies?, in: G. Brewka, J. Lang (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008, AAAI Press, 2008, pp. 285–295.
URL <http://www.aaai.org/Library/KR/2008/kr08-028.php>
- [35] S. Ghilardi, C. Lutz, F. Wolter, Did I damage my ontology? A case for conservative extensions in description logics, in: P. Doherty, J. Mylopoulos, C. Welty (Eds.), Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR'06), AAAI Press, 2006, pp. 187–197.
- [36] C. Lutz, D. Walther, F. Wolter, Conservative extensions in expressive description logics, in: M. M. Veloso (Ed.), IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007, 2007, pp. 453–458.
URL <http://dli.iiit.ac.in/ijcai/IJCAI-2007/PDF/IJCAI07-071.pdf>

- [37] F. Martin-Recuerda, D. Walther, Axiom dependency hypergraphs for fast modularisation and atomic decomposition, in: M. Bienvenu, M. Ortiz, R. Rosati, M. Simkus (Eds.), Proceedings of the 27th International Workshop on Description Logics (DL'14), Vol. 1193 of CEUR Workshop Proceedings, 2014, pp. 299–310.
- [38] B. C. Grau, I. Horrocks, Y. Kazakov, U. Sattler, Modular reuse of ontologies: Theory and practice, *J. Artif. Intell. Res. (JAIR)* 31 (2008) 273–318. doi:10.1613/jair.2375.
URL <http://dx.doi.org/10.1613/jair.2375>
- [39] A. Artale, D. Calvanese, R. Kontchakov, M. Zakharyashev, The DL-lite family and relations, *J. Artif. Intell. Res. (JAIR)* 36 (2009) 1–69.
- [40] F. Baader, S. Brandt, C. Lutz, Pushing the EL envelope, in: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence, IJCAI-05, Professional Book Center, 2005, pp. 364–369.
- [41] E. Sandewall, Defeasible inheritance with doubt index and its axiomatic characterization, *Artif. Intell.* 174 (18) (2010) 1431–1459.
- [42] F. S. Y. Kazakov, M. Krotzsch, ELK reasoner: Architecture and evaluation, in: E. J.-R. I. Horrocks, M. Yatskevich (Ed.), Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE-2012), CEUR Workshop Proceedings, 2012.
- [43] Y. Kazakov, M. Krötzsch, F. Simancik, The incredible ELK - from polynomial procedures to efficient reasoning with ontologies, *J. Autom. Reason.* 53 (1) (2014) 1–61. doi:10.1007/s10817-013-9296-3.
URL <https://doi.org/10.1007/s10817-013-9296-3>
- [44] Y. Kazakov, P. Klinov, Incremental reasoning in EL+ without bookkeeping, in: Description Logics, 2013, pp. 294–315.
- [45] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, Z. Wang, Hermit: An OWL 2 reasoner, *Journal of Automated Reasoning* 53 (3) (2014) 245–269. doi:10.1007/s10817-014-9305-1.
URL <http://dx.doi.org/10.1007/s10817-014-9305-1>
- [46] A. A. Romero, M. Kaminski, B. C. Grau, I. Horrocks, Ontology module extraction via Datalog reasoning, in: B. Bonet, S. Koenig (Eds.), Proceedings

of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA, AAAI Press, 2015, pp. 1410–1416.
URL <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9842>

- [47] G. Casini, T. Meyer, K. Moodley, R. Nortje, Relevant closure: A new form of defeasible reasoning for description logics, in: E. Fermé, J. Leite (Eds.), Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings, Vol. 8761 of Lecture Notes in Computer Science, Springer, 2014, pp. 92–106. doi:10.1007/978-3-319-11558-0_7.
URL https://doi.org/10.1007/978-3-319-11558-0_7
- [48] L. Giordano, V. Gliozzi, G. L. Pozzato, R. Renzulli, An efficient reasoner for description logics of typicality and rational closure, in: Proceedings of the 30th International Workshop on Description Logics, Montpellier, France, July 18-21, 2017, 2017.
- [49] P. Ferraris, J. Lee, V. Lifschitz, Stable models and circumscription, *Artif. Intell.* 175 (1) (2011) 236–263. doi:10.1016/j.artint.2010.04.011.
URL <https://doi.org/10.1016/j.artint.2010.04.011>
- [50] G. Gottlob, A. Hernich, C. Kupke, T. Lukasiewicz, Equality-friendly well-founded semantics and applications to description logics, in: J. Hoffmann, B. Selman (Eds.), Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada, AAAI Press, 2012.
URL <http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/5100>
- [51] A. Hernich, C. Kupke, T. Lukasiewicz, G. Gottlob, Well-founded semantics for extended datalog and ontological reasoning, in: R. Hull, W. Fan (Eds.), Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2013, New York, NY, USA - June 22 - 27, 2013, ACM, 2013, pp. 225–236. doi:10.1145/2463664.2465229.
URL <https://doi.org/10.1145/2463664.2465229>
- [52] G. Gottlob, A. Hernich, C. Kupke, T. Lukasiewicz, Stable model semantics for guarded existential rules and description logics, in: C. Baral, G. D. Giacomo, T. Eiter (Eds.), Principles of Knowledge Representation and

Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014, AAAI Press, 2014.

URL <http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/8011>

- [53] M. Alviano, A. Pieris, Default negation for non-guarded existential rules, in: T. Milo, D. Calvanese (Eds.), Proceedings of the 34th ACM Symposium on Principles of Database Systems, PODS 2015, Melbourne, Victoria, Australia, May 31 - June 4, 2015, ACM, 2015, pp. 79–90. doi:10.1145/2745754.2745758.
URL <https://doi.org/10.1145/2745754.2745758>
- [54] M. Alviano, M. Morak, A. Pieris, Stable model semantics for tuple-generating dependencies revisited, in: E. Sallinger, J. V. den Bussche, F. Geerts (Eds.), Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017, ACM, 2017, pp. 377–388. doi:10.1145/3034786.3034794.
URL <https://doi.org/10.1145/3034786.3034794>
- [55] H. Wan, G. Xiao, C. Wang, X. Liu, J. Chen, Z. Wang, Query answering with guarded existential rules under stable model semantics, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, AAAI Press, 2020, pp. 3017–3024.
URL <https://aaai.org/ojs/index.php/AAAI/article/view/5695>
- [56] M. Alviano, N. Leone, M. Manna, G. Terracina, P. Veltri, Magic-sets for datalog with existential quantifiers, in: P. Barceló, R. Pichler (Eds.), Datalog in Academia and Industry - Second International Workshop, Datalog 2.0, Vienna, Austria, September 11-13, 2012. Proceedings, Vol. 7494 of Lecture Notes in Computer Science, Springer, 2012, pp. 31–43. doi:10.1007/978-3-642-32925-8_5.
URL https://doi.org/10.1007/978-3-642-32925-8_5
- [57] M. Alviano, N. Leone, P. Veltri, J. Zangari, Enhancing magic sets with an application to ontological reasoning, Theory Pract. Log. Program. 19 (5-6)

(2019) 654–670. doi:10.1017/S1471068419000115.

URL <https://doi.org/10.1017/S1471068419000115>